PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 7 : G06F 17/30, 17/24

A1

(11) International Publication Number: **WO 00/39713**

(43) International Publication Date: 6 July 2000 (06.07.00)

(21) International Application Number: PCT/US99/30965

(22) International Filing Date: 23 December 1999 (23.12.99)

(30) Priority Data:

60/114,065	28 December 1998 (28.12.98)	US
60/160,639	20 October 1999 (20.10.99)	US

(71) Applicant: GEMTEQ SOFTWARE, INC. [US/US]; Suite 206, 1450 Grant Avenue, Novato, CA 94945 (US).

(72) Inventors: GULATI, Ashwin; 204 Redding Way, San Rafael, CA 94901 (US). BLACKBURN, William, J.; 488 Oak Manor Drive, Fairfax, CA 94930 (US).

(74) Agents: DUBORD, Renee et al.; Fenwick & West LLP, Two Palo Alto Square, Palo Alto, CA 94306 (US).

(81) Designated States: AE, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, TJ, TM, TR, TT, UA, UG, UZ, VN, YU, ZA, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published

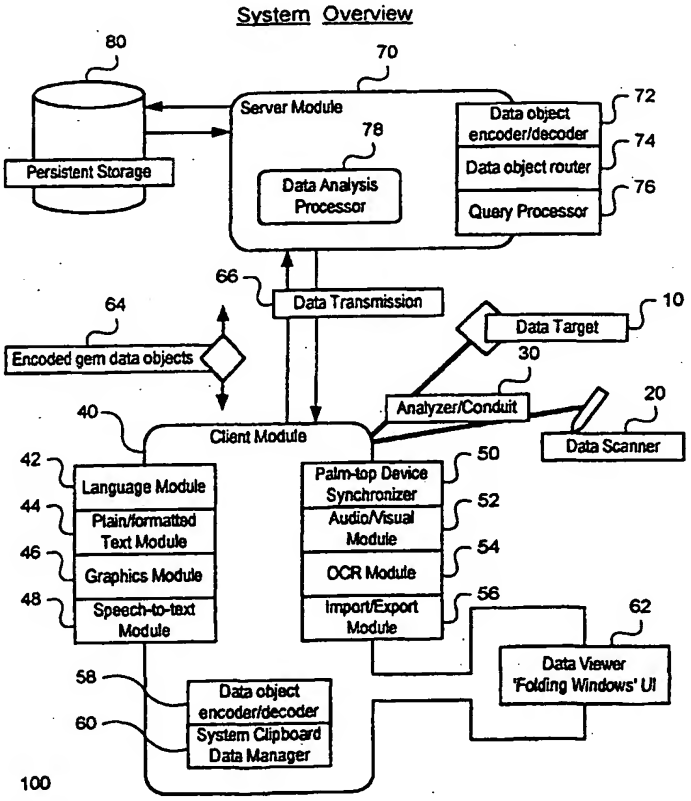
With international search report.

Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.

(54) Title: A METHOD AND SYSTEM FOR PERFORMING ELECTRONIC DATA-GATHERING ACROSS MULTIPLE DATA SOURCES

(57) Abstract

A method and system for electronic data-gathering system allows a user to easily capture and archive electronic data without the need to interact with an additional application user-interface. The present invention streamlines the workflow of performing research, and also ensures that information is easily traceable to its original source. The described embodiments of the present invention automatically encapsulate user-selected sets of electronic data with a set of attribution, creation, and user-defined metadata. The system uses the captured data and metadata to create gem data objects. These gem data objects are then routed within the electronic data-gathering system. The gem data objects may be stored on a persistent storage mechanism. The research system also includes a data viewer that allows a user to view and perform actions upon the gem data objects.



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	**ES**	Spain	**LS**	Lesotho	**SI**	Slovenia
AM	Armenia	**FI**	Finland	**LT**	Lithuania	**SK**	Slovakia
AT	Austria	**FR**	France	**LU**	Luxembourg	**SN**	Senegal
AU	Australia	**GA**	Gabon	**LV**	Latvia	**SZ**	Swaziland
AZ	Azerbaijan	**GB**	United Kingdom	**MC**	Monaco	**TD**	Chad
BA	Bosnia and Herzegovina	**GE**	Georgia	**MD**	Republic of Moldova	**TG**	Togo
BB	Barbados	**GH**	Ghana	**MG**	Madagascar	**TJ**	Tajikistan
BE	Belgium	**GN**	Guinea	**MK**	The former Yugoslav Republic of Macedonia	**TM**	Turkmenistan
BF	Burkina Faso	**GR**	Greece			**TR**	Turkey
BG	Bulgaria	**HU**	Hungary	**ML**	Mali	**TT**	Trinidad and Tobago
BJ	Benin	**IE**	Ireland	**MN**	Mongolia	**UA**	Ukraine
BR	Brazil	**IL**	Israel	**MR**	Mauritania	**UG**	Uganda
BY	Belarus	**IS**	Iceland	**MW**	Malawi	**US**	United States of America
CA	Canada	**IT**	Italy	**MX**	Mexico	**UZ**	Uzbekistan
CF	Central African Republic	**JP**	Japan	**NE**	Niger	**VN**	Viet Nam
CG	Congo	**KE**	Kenya	**NL**	Netherlands	**YU**	Yugoslavia
CH	Switzerland	**KG**	Kyrgyzstan	**NO**	Norway	**ZW**	Zimbabwe
CI	Côte d'Ivoire	**KP**	Democratic People's Republic of Korea	**NZ**	New Zealand		
CM	Cameroon			**PL**	Poland		
CN	China	**KR**	Republic of Korea	**PT**	Portugal		
CU	Cuba	**KZ**	Kazakstan	**RO**	Romania		
CZ	Czech Republic	**LC**	Saint Lucia	**RU**	Russian Federation		
DE	Germany	**LI**	Liechtenstein	**SD**	Sudan		
DK	Denmark	**LK**	Sri Lanka	**SE**	Sweden		
EE	Estonia	**LR**	Liberia	**SG**	Singapore		

A METHOD AND SYSTEM FOR PERFORMING ELECTRONIC DATA-GATHERING ACROSS MULTIPLE DATA SOURCES

INVENTORS

Ashwin Gulati

William J. Blackburn

CROSS REFERENCE TO RELATED APPLICATION

This application claims the benefit of U.S. Provisional Patent Application Serial No. 60/114,065, "Virtual Gem", by Ashwin Gulati, filed December 28, 1998, and U.S. Provisional Patent Application Serial No. 60/160,639, "Method And System For Performing Electronic Data-Gathering Across Multiple Data Sources", by Ashwin Gulati and William J. Blackburn, filed October 20, 1999, which subject matter is incorporated herein by reference.

BACKGROUND

A. Technical Field

The present invention relates generally to electronic data-gathering, and more particularly, to the automatic capture, storage and classification of selected portions of electronic data.

B. Background of the Invention

Researching a topic using digital sources has become a time intensive process due to the vast quantity of data available. In addition, a researcher needs to track and attribute all such data used to its original source. Digital information may be collected from such varied sources as text and image files, network sources, and the Internet. The current process of collecting such data involves a complex series of interactions with the source data itself, in addition to computer-based or manual filing systems, word processors, and other authoring tools and applications that make the process non-linear and difficult to manage.

Researchers using electronic media as sources for information and documentation can archive research data using file systems or indexing software solutions. The traditional method for storing a copy of information found during research is to save the entire document into such a file system for later retrieval, or alternatively to save a relevant section of a

document in a word processing or authoring type of computer application. These types of traditional computer applications for research storage include an underlying model describing the nature of the persistent data used and the functions that are available to operate on that data. The computer system will implement one or more views, typified in the modern GUI interface, allowing the user to interact with the model.

For example, a user might want to save a selected portion of information from a web page viewed using a web browser. The user could save the complete web page; however, this would not only save the relevant information but would store the entire web page. Storing the entire web page would distract future users from focusing upon the relevant information within the web page. Such storing of extraneous information is undesirable in a research environment.

The user could alternatively copy the relevant section of information from the web browser view. Next, the user would open up another application view, such as a word processing application, and paste the document segment into the application. The user would then have to save the new document as a new file. The user would also have to manually input creation and attribution information about the document into either the document itself or the file name of the document, so that the user could properly identify and attribute the information later. As used herein, creation information refers to information about the actions relating to saving the original data, such as the identity of the system user, the date and time of storage, and the source document from which the data was taken. Attribution information refers to bibliographic information such as the original author, date of publication, etc.

Currently, multiple pieces of stored electronic data are most often viewed using the application with which they were stored. A user might store entire web pages, portions of text, spreadsheet information, and graphics. Each of these types of documents could potentially be stored in a different application format. When each piece of data is later retrieved, it is necessary to open multiple different applications for data viewing, adding complexity to the process.

Traditional approaches to electronic data-gathering interrupt the workflow of a research project. For example, to save a current view, a researcher must stop interacting with the current view of the program he is using to perform his research; interact with a view of the file system browser, indexing application, or word processing application required to view

stored research information; and then return to the view he was accessing before. The preferred process would be to organize such research efforts into a stream where all relevant research materials are collected from source documents at the beginning of the project. These collected materials would be processed to automatically retain creation and attribution information about the materials, and then stored for future review. Furthermore, the data stored in this phase would be visible to other researchers working on the same, or even non-related projects. Finally, the data could be withdrawn as needed, along with all proper creation and attribution information, during the compilation of the final output document.

No conventional system allows for the automatic capture, storage, and classification of less than an entire source document, including the acquisition and storage of creation and attribution metadata, packaged into a single, routable self-attributing format. Thus, there is a need for a system that will allow a researcher to acquire and use important pieces of data gleaned from electronic files of various types without stopping his or her work to interact with an additional application user-interface. Such a system would provide a conduit for inserting research data into a system model directly while negating or delaying the need to interact with a view of the system.

SUMMARY OF THE INVENTION

The electronic data-gathering system of the present invention allows a user to easily capture and archive electronic data without the need to interact with an additional application user-interface. This streamlines the workflow of performing research, and also ensures that information is easily traceable to its original source.

The described embodiments of the present invention automatically encapsulate user-selected sets of electronic data with a set of attribution, creation, and user-defined metadata. The system uses the captured data and metadata to create gem data objects. These gem data objects are then routed within the electronic data-gathering system. The gem data objects may be stored on a persistent storage mechanism, and viewed or edited by a system user.

In one embodiment of the present invention, a user captures a set of electronic data via the system clipboard, drag and drop activity, manual type-in or a scanner. The data is input into a data target, which does not necessitate opening an additional program view. However, the user may optionally choose to input data using a system data viewer. The research system automatically captures a set of metadata and creates an interim object by appending the set of

metadata to the original captured data. This interim object is then passed to the main routine, where a gem data object is created.

The research system also includes a data viewer that allows a user to perform actions upon the gem data objects. A user may perform a search for gem data objects using keywords or other attributes of gem data objects. A user may view the gem object database in a hierarchical manner, and create a storage hierarchy by placing gem data objects in various containers within the database. Additionally, a user may edit gem data objects using the data viewer.

Advantages of the invention will be set forth in part in the description which follows and in part will be obvious from the description or may be learned by practice of the invention. The objects and advantages of the invention will be realized and attained by means of the elements and combinations particularly pointed out in the appended claims and equivalents.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is an example of an electronic data-gathering system in accordance with the present invention.

Fig. 2 is a flowchart of a research workflow method of the present invention.

Fig. 3 is a flowchart showing an example of a gathering phase for a research workflow method.

Fig. 4A is an illustration of an acquire data step using a data target as used in an embodiment of the present invention.

Fig. 4B is an illustration of an acquire data step using a gem data object graphical user interface view as used in an embodiment of the present invention.

Fig. 5 is flowchart showing an example of a method for processing data into a gem data object.

Fig. 6 is a flowchart of an example of a method for creating an interim object.

Fig. 7 is a flowchart of an example of a method for creating a gem data object.

Fig. 8 is a diagram showing an example of a user interface window for a gem data object viewer with the "General" tab of the popup menu of a gem data object expanded.

Fig. 9A is a diagram showing an example of a user interface window for a gem data object viewer with the "Bibliography" tab of the popup menu of a gem data object expanded.

Fig. 9B is a diagram showing an example of the "Bibliography" tab of the popup menu of a gem data object expanded, with a scrolling menu displayed for the "Bibliography Format" option.

Fig. 10 is a diagram showing an example of a user interface window for a gem data object viewer with the "View" tab of the popup menu of a gem data object expanded.

Fig. 11 is a flowchart showing an example of a storage phase for a research workflow method.

Fig. 12 is a flowchart showing an example of a performing data actions phase for a research workflow method.

Fig. 13 is a flowchart showing an example of a data analysis process.

Fig. 14 is an example of a root node as used in an embodiment of the present invention.

Fig. 15 is an example of a container node as used in an embodiment of the present invention.

Fig. 16 is an example of a container node as used in an embodiment of the present invention.

Fig. 17 is an example of a gem data object as used in an embodiment of the present invention.

Fig. 18 is a diagram showing an example of a gem data object viewer with a "View" popup menu expanded to show a text type gem data object.

Fig. 19 is a diagram showing an example of a gem data object viewer with a "View" popup menu expanded to show a text type gem data object.

Fig. 20 is a diagram showing an example of a gem data object viewer with a "View" popup menu expanded to show a file type gem data object.

Fig. 21 is a diagram showing an example of a gem data object viewer with a "View" popup menu expanded to show a graphics type gem data object.

Fig. 22 is a diagram showing an example of a gem data object viewer with a "View" popup menu expanded to show a Uniform Resource Locator (URL) type gem data object.

DETAILED DESCRIPTION OF EMBODIMENTS

Reference will now be made in detail to several embodiments of the present invention, examples of which are illustrated in the accompanying drawings. Wherever practicable, the same reference numbers will be used throughout the drawings to refer to the same or like parts.

A. System Overview

Fig. 1 is an illustration of an embodiment of an electronic data-gathering system for use in research operations. System 100 is used for creating, viewing, storing and using encapsulated packages of electronic data and metadata. These encapsulated packages of data and metadata are referred to herein as gem data objects. It should be understood that the term "gem data object" refers not only to a Gem from the eGems ™ software by Gemteq Software, Inc. but also to other types of objects. Different formats for creating, storing, and using gem data objects will be evident to one of skill in the art.

The system of Fig. 1 is suitable for use with both data acquisition and data retrieval. First, the data acquisition functions will be discussed, followed by the data retrieval functions.

The data acquisition functions of the electronic data-gathering system of Fig. 1 will be presented by broadly tracing the path of a set of data that a user wishes to enter into the system. First, electronic data must have a point of entry into a system 100. The system 100 contains a data target 10 through which electronic data may be captured by the system 100. Alternatively, electronic data may be captured using a data scanner 20. The captured data is then passed to an analyzer/conduit 30, which receives the captured data, captures additional metadata, and passes the data and metadata to a client module 40. The metadata captured by the analyzer/conduit 30 may be any data describing or referring to the original data.

The client module 40 now contains both the original data and the captured metadata. Within the client module 40, different processing modules operate on the data and metadata to create a gem data object, an encapsulated package of data and metadata, capable of being routed throughout the system 100. Specialized modules within the client module 40 operate on the gem data object and further enhance data acquisition, storage and usage. A plain/formatted text module 44, a graphics module 46, and an import/export module 56 make

up the core functionality for acquiring and using basic image and text data, as well as packaging that data for transfer between databases.

The text module 44 is responsible for processing acquired text data sets. Text data sets may be defined using a system clipboard, drag and drop activity, manual typing-in of data, speech recognition of audio data, or scanning-in of data combined with optical character recognition (OCR). Processing text data includes recognizing the type of text, displaying it in a visual component, and performing editing operations upon it.

The graphics module 46 is responsible for processing acquired image data sets. Image data sets may be defined using a system clipboard, drag and drop activity, or scanning-in of data. Processing includes recognizing the type of image, displaying it in a visual component, and performing editing operations upon it.

The import/export module 56 handles the transfer of data between databases. In one embodiment, this module resides in the client module 40, but it may also reside in the server module 70. The import/export module 56 provides for the transfer of all or part of the current system database into a new or existing database or export file, as well as the transfer of some or all of a database or import file into the current system database.

A language module 42 is capable of performing in-line language translation upon the captured data to convert from text in one language to text in another language. A speech-to-text module 48 allows for the translation of text data into audio data and vice versa. A palm-top device synchronizer module 50 allows for the acquisition of data from palm-top devices during routine synchronization procedures, which such palm-top devices are capable of performing in conjunction with another computer. An audio/visual module 52 allows for the capture, storage and retransmission of audio and video data. An optical character recognition (OCR) module 54 allows for the conversion of scanned text into editable text data. The specialized client modules shown in Fig. 1 are merely representative. Additional modules to perform specialized manipulations for gem data objects may be added to the client module 40, as will be evident to one of skill in the art.

A data viewer 62 allows the user to interact with the system 100 to manipulate raw electronic data and gem data objects in several ways. Such user interactions include, but are not limited to: capturing original data; gathering, refining, and editing gem data objects; classifying and storing gem data objects; and retrieving and using gem data objects. In one

embodiment, the data viewer 62 provides a graphical user interface that displays all gem data object database information in a hierarchical format.

A data object encoder/decoder 58 packs information from the client module 40 for transporting to a server module 70. During data acquisition, the data object encoder/decoder 58 will package the newly created gem data object into a format suitable for transport. The data object encoder/decoder 58 also unpacks information received back from the server module 70. A system clipboard data manager 60 allows a user to cut and paste data. The clipboard data manager 60 handles interpreting a selected gem data object during copy and paste or drag and drop actions, and makes the selected data available to the requesting application.

After the creation and encoding of a gem data object within the client module 40, encoded gem data object 64 is sent to a data transmission module 66 for transport to the server module 70. The specific implementation of the data transmission module 66 will vary based upon the type of client/server architecture within the system 100. For example, for a local, single user installation, no transmission protocol is required, as data transmission may be done using standard in-process communication mechanisms. For a remote server module, some means of requesting data, or operations on data, and for receiving a response is required. The data transmission module 66 can therefore use any number of different data transmission protocols, which are well known in the art.

The server module 70 receives the encoded gem data objects from the data transmission module 66. The server module 70 encapsulates all data access to an underlying persistent storage mechanism 80. The server module 70 can run in or out of process with the client module 40, and can run either locally, as would be the case with a single user version of the system, or remotely, as in a client-server implementation or over the Internet, via standard protocols.

A data object encoder/decoder 72 unpacks the transmission, in this example a newly created gem data object, received from the client module 40. The server module 70 passes the transmission to a data object router 74, which processes client requests and routes data to the appropriate gem object database location. The data object router 74 may route data based upon routing information received with a transmission, or may route according to pre-set rules.

The server module 70 interacts with the persistent storage 80 to persist the data. The newly created gem data object will thus be stored in the persistent storage 80 in a gem object database. Various different embodiments of a gem object database will be evident to one of skill in the art.

The system 100 of Fig. 1 may also be used for data retrieval. Data retrieval may include, but is not limited to, retrieving individual gem data objects and performing queries for information upon the gem object database.

When using the system 100 for data retrieval, a user will input requests for information into the data viewer 62. The data viewer 62 provides a graphical user interface for viewing the contents of the gem object database and performing searches of the gem object database. In addition, the data viewer 62 provides standard graphical means (such as drag and drop or copy and paste) for organizing data via the movement, addition, and deletion of categories and gem data objects.

The client module 40 receives the request from the data viewer 62. The data object encoder/decoder 58 packages the request for transmission to the server module 70. The encoded request is transmitted to the server module 70 via the data transmission module 66. The server module 70 receives the request from the data transmission module 66, decodes it using the data object encoder/decoder 72, and submits it to a query processor 76.

The query processor 76 forms database queries based upon requests received from the client module 40 or generated locally within the server module 70. These queries are used to search the gem object database and respond to requests for database searches. The response is then encoded via the data object encoder/decoder 72 and retransmitted back to the client module 40. The client module 40 will display the request results to the user via the data viewer 62.

The query processor 76 and a data analysis processor 78 perform additional server module 70 tasks. The data analysis processor 78 analyzes the gem object database stored within the system 100 to create general profiles of system 100 users, and to create specific profiles of topics of interest within the gem object database. The data derived from these profiles may then be formed into queries. These queries may be submitted to Internet search engines as well as to the query processor 76 to provide a user with Internet sites or local files of potential interest.

The various modules disclosed with respect to Fig. 1 are merely representative. The functionality of the various different modules may be combined into a non-modular electronic data-gathering system without departing from the inventive concepts disclosed herein. Various different modules may also be combined into a single module as will be evident to one of skill in the art.

The embodiment disclosed above is implemented in software. The software is embodied in a computer readable medium, suitable for use with a computer system. Such a computer system may comprise a plurality of processors combined with a data viewscreen; however, other embodiments will be evident to one of skill in the art.

Fig. 2 shows a flowchart of a research workflow method of the present invention. The method of Fig. 2 is used with the electronic data-gathering system 100 of Fig. 1 to perform research operations. During a gathering phase 210, important and relevant information is chosen and collected. Next, the information is stored in a storage phase 220. Finally, the information may be retrieved, used, and manipulated in a performing data actions phase 230. Although the process shown here is a linear one, the steps can be performed in any order. For example, a user could perform data actions upon the data before storage.

B. Gathering Phase

Fig. 3 shows further details of an embodiment of the gathering phase of the research workflow method for use with an electronic data-gathering system. Data is first acquired in step 310, and then data is processed into a gem data object in step 320.

Figs. 4A and 4B show two different embodiments of the acquire data step. Four different methods of selecting data are shown: interaction with the system clipboard 410; drag and drop activity 420; manual typing-in of data 430; and electronic scanning-in of data 440. A system user may capture data electronically using any of these different methods. A user may also alternate between the different methods for selecting data as convenience dictates.

A user may capture any kind of electronic data in the acquire data step. The list of potential types of data that may be selected includes, but is not limited to: text type data, including Rich Text Format (RTF), American Standard Code for Information Interchange (ASCII) and various word-processing native formats; graphical files including Joint Photographic Expert Group (JPEG) and Graphics Interchange Format (GIF) type files; URL

links; World Wide Web (WWW) pages, including Hyper Text Markup Language (HTML) files; File links; and Object Linking and Embedding (OLE) type object links.

Different types of gem data objects may be created from the data types listed above. In addition, gem data objects may be created from other types of data as will be evident to one of skill in the art. Support for additional gem data object formats may be included by coding and installing the appropriate module. There is no limit on the type and number of potentially supported formats except for the disk space and memory of the hardware platform chosen by the user.

In Fig. 4A, these four different methods of selecting data 410, 420, 430 and 440 are used to input data into a data target 400. The data target 400 is a free-floating, movable icon symbolizing the active research system. The data target 400 eliminates the need for a user to open and interact with additional application windows, while still allowing the user to access key features of the system. As shown in Fig. 4A, electronic data may be input directly into the data target without the need for an additional view window. In this case, the data being input is either pasted or dragged onto the data target 400. The data target 400 provides a target for inputting data into the research system, and also provides a visual indication that the electronic data-gathering system is running. The user can add data to any level of the tree structure of the research system. If data is added to data target 400, the data is placed at a default location in the tree.

In Fig. 4B, the aforementioned methods of selecting data 410, 420, 430 and 440 are used to input data into a data viewer window 450. In this embodiment, the data target 400 has been expanded to allow a view into the research system. The data viewer window 450 provides a user with an expanded view of the research system, while still allowing a user to work concurrently with other program applications.

Fig. 5 shows an embodiment of the method for processing the selected raw data into a gem data object. In one embodiment, the steps of acquiring metadata 510 and creating an interim object 520 are performed by the analyzer/conduit 30. However, other embodiments will be evident to one of skill in the art. For example, the metadata could be passed directly to the main routine without creating an interim object. The main processing thread of either the client module 40 or the server module 70 is referred to herein as the main routine.

First, metadata is acquired in step 510. This metadata is information about the original data obtained in the gathering phase. Such metadata may include, but is not limited to, information about the source of the original data, such as: a Web page address where the original data was found; the filename of the source of the data; or the date the data was copied. The metadata may also include attribution data required to format a proper bibliography, such as: the original author; date of publication; or specific location or page number where the original data was located within the source document.

The metadata is used to create an interim object in step 520. This interim object is then passed to the main routine via a capture event 530. The main routine is then used in step 540 to create a gem data object from the interim object.

Fig. 6 shows an embodiment of the method of operation for the analyzer/conduit 30 to create an interim object. The steps shown in Fig. 6 encompass steps 510 and 520 from Fig. 5. Steps 610-660 all relate to acquiring metadata and may be performed in any order. Not all steps will be performed in all embodiments.

In step 610, user data is added to the set of metadata being collected. The operating system (OS) Application Program Interface (API) is used to acquire the user data, which may include: information about the system user who is collecting data; the name or designation of the machine being used; and the system date and time. In step 620, source data is added to the metadata. The OS API is used to determine the source application for the original data as well as the designation of the document providing the original data. For example, if the original data was a web page, the source application might be a web browser while the document designation would be a URL. However, in certain situations this information may not be available. In such cases, a best-guess routine is used which attempts to determine the application name and document name providing the data. The routine makes a guess based upon the application window that was last active before the user input data into the electronic data-gathering system.

Bibliographic data is added in step 630. The captured data is scanned to determine such information as the author, the publication date, the original document title, and other attribution data. When this data is not available, a best-guess routine is used which suggests a value from the captured data, for example "meta" tags in HTML files, or embedded tags in

RTF files. Optionally, the original source data may also be scanned to determine a best guess for this information. If no data is available to make a best guess, the field is left blank.

A default name and description is added in step 640. The captured data is scanned to suggest a default name for the gem data object that will be created. This name is typically derived from the first 3-4 words of captured text, but other implementations will be evident to one of skill in the art. The description will default to "<data type> from <source document>."

A keyword scan is performed in step 650. Any text associated with the captured data is scanned and separated into individual words. A dictionary of "small words" is applied to discard words that do not make meaningful keywords. The resulting collection of keywords may be used for multiple purposes, such as suggesting related search terms or routing the resulting gem data object after it is created.

The data is scanned for embedded URLs in step 660. Any embedded URLs found are stored and may be used for multiple purposes, such as suggesting related web links or routing the resulting gem data object after it is created.

Interim object packing is performed in step 670. The collection of captured data and metadata is packaged into an interim object that may be passed to the main routine for further processing. Not all embodiments perform this step.

Fig. 6 shows only one embodiment of the various types of metadata that may be captured by the electronic data-gathering system. Many additional types of metadata may be captured and encapsulated along with the original captured data, as will be evident to one of skill in the art. In one embodiment, metadata is initially captured automatically by the electronic data-gathering system. However, a user may also add metadata to a gem data object as well. The user of the data-gathering system may also decide not to capture certain types of metadata disclosed herein without departing from the scope of the present invention.

Fig. 7 is a flowchart of a method to create a gem data object from captured data and metadata. The steps shown in Fig. 7 encompass step 540 of Fig. 5. The steps of Fig. 7 are performed by the main routine.

In step 710, the analyzer/conduit 30 notifies the main routine of available new data via an event notice. This prompts the main routine to retrieve the interim object in step 720. The main routine also retrieves an empty gem data object from the server module in step 730. The

main routine examines the captured data in order to recognize the data type, and applies the client module or modules corresponding to that data type to the captured data in step 740. In step 750 the main routine populates the fields of the empty gem data object with the data from the interim object.

The resulting gem data object is an implementation of the encapsulated original data and metadata along with the functionality required to use, view, and manipulate the captured data. The gem data object is a single shareable package of data and metadata. This gem data object package is a "self-attributing" unit - it contains bibliographic information as well as information about how an appropriate bibliography, footnote, or caption should be formatted. This enables the gem data object to generate its own bibliography or other identifying caption about itself.

Figs. 8, 9, and 10 are diagrams showing an embodiment of a user interface for the electronic data-gathering research system. The user interface provides one or more views into the research system model and displays information about the gem data objects within the system. The views also provide a user interface for adding, modifying, or deleting the original data or metadata encapsulated by the gem data object. The windows shown in Figs. 8, 9, and 10 are merely representative of the type of information that may be displayed to describe a gem data object.

In Fig. 8, a window 810 shows a "Gem View" window providing a hierarchical view of the stored gem data objects within the system. The individual gem data objects may be organized in a hierarchical manner by grouping them into containers. Each container is a grouping of individual gem data objects, or a grouping of containers. This hierarchical storage method provides a convenient way for a user to organize the gem data objects within the research system. For example, in the window 810, "PTO Website" constitutes a container 812, while "Unfiled" constitutes another container 816. A gem data object "Functions of Patent and Trademark Office" 814 is stored in the container 812, while a gem data object "Future paper" 818 is stored in the container 816. In this example, gem data object "Functions of the Patent and Trademark Office" 814 has a data type of "text".

A window 820 shows an expanded view of a popup window for the gem data object "Functions of Patent and Trademark Office" 814. A "General" tab 822 of the popup window 820 has been selected. The tab 822 displays information regarding the gem data object's name

and descriptive notes regarding the gem data object. The tab 822 also displays the container in which the gem data object is located, as well as information about the original source application for the data contained in the gem data object. Tab 822 provides a user interface for viewing, adding, deleting, or modifying the displayed information regarding a gem data object.

Fig. 9A contains the same window 810 showing a "Gem View" window providing a hierarchical view of the stored gem data objects within the system as shown in Fig. 8. However, in Fig. 9A a popup window 920 for the gem data object 814 has a "Bibliography" tab 922 selected. The tab 922 displays bibliographic information regarding the original source data, such as: the author's name; publication; the place published; the edition or volume; and the page numbers. The tab 922 also displays the original article title and date of publication, as well as the URL of the source document. Tab 922 provides a user interface for adding, deleting, or modifying the displayed information regarding a gem data object.

Fig. 9B shows a second view of the popup window 920 with the "Bibliography" tab 922 selected. In Fig. 9B, a scrolling menu 924 is displayed for the "Bibliography Format" option. The scrolling menu 924 allows a user to select from various different types of standard bibliographic formats for citing printed and electronic resources. The bibliography format field of the gem data object contains an integer value code representing the chosen type of bibliography format.

The bibliography entry itself is composed "on-the-fly" when requested by the user. The user can request the insertion of a bibliographic entry, footnote, or other type of caption by using drag and drop or copy and paste interactions. Once requested, a method of the gem data object is invoked wherein the format code is evaluated. The data fields corresponding to source information (such as author, publication, etc.) are then strung together using the appropriate formatting (such as italics, quotation marks, etc.) and made available to the target application via the system clipboard.

Fig. 10 contains the same window 810 showing a "Gem View" window providing a hierarchical view of the stored gem data objects within the system as shown in Figs. 8 and 9A. However, in Fig. 10 a popup window 1020 for the gem data object 814 has a "View" tab 1022 selected. The tab 1022 provides a view of the original captured data. Tab 1022 also provides a user interface for adding, deleting, or modifying the data contained in the gem data object.

In the example shown, the user can perform standard editing operations such as changing the font, cutting and pasting, justifying the text, etc.

C. Storage Phase

After a gem data object is created in the gathering phase, it is stored so that it can be used in future research. Multiple different types of processes may be performed using the accumulated stored data. For example, individual gem data objects may be retrieved, the storage database can be reordered or reorganized, and the storage database can be searched. The actual type of persistent storage system used is masked from the underlying workings of the system by the server component. The role of persistent storage may thus be implemented in any of a number of commercially available database systems, as will be evident to one of skill in the art.

Fig. 11 is a flowchart of a storage phase for a research workflow method. Fig. 11 relates to the initial storage of a new gem data object through interaction between the client module 40, the server module 70, and the persistent storage module 80 as shown in Fig. 1. Various different types of client/server architecture may be used in the research workflow method of the present invention. For example, in a single-user version of the system, the server module could be local to the client module. Alternatively, a more distributed client/server system could be used wherein the server module is remote from the client module. Although specific examples of client/server architectures will be presented herein, the present invention encompasses all of the various types of client/server architectures suitable for use with the present invention.

In the storage phase of Fig. 11, the gem data object is packaged by the client module 40 for transmission to the server module 70. Optionally, the raw data comprising the gem data object may first be compressed in step 1112. Next, the gem data object is encoded in step 1114, where it is converted into a form suitable for transfer over existing network protocols. In one embodiment, eXtensible Markup Language (XML) is used to encode the gem data object and an XML-RPC (Remote Procedure Call) is used to wrap the gem data object in a method call that provides the data and any additional required parameters to the server module 70. For example, the XML-based implementation would contain a method call requesting that the gem data object be saved, information about the database in which to save it, and other required parameters.

The encoded data is then transmitted in step 1120 from the client module 40 to the server module 70. In an XML-based embodiment, the client module 40 contacts the server module 70 on a well-known port using Hyper Text Transfer Protocol (HTTP) and Java Servlets, and a socket is opened. Upon establishing a successful connection, the client module 40 transmits the XML-RPC method call containing the XML-encoded gem data object.

The server module 70 then decodes the gem data object in step 1132, and decompresses the gem data object in step 1134 if the gem data object was compressed in step 1112. In an XML-based embodiment, the XML tags are parsed and the transmitted gem data object is rebuilt and forwarded for routing, along with passed parameters such as the name of the requested storage database.

In step 1136, the gem data object is routed to a storage location. The transmitted gem data object may already have a pre-assigned database location. If so, this information is passed along with the gem data object and used for routing purposes.

If location or other required information is missing, the server module 70 can apply pre-set rules that have been configured on the server module 70 to the gem data object. Such pre-set rules can be developed for routing based upon any type of data or metadata contained within the gem data object. For example, if a destination location within the database is not supplied, a rule might exist which routes all data containing a particular phrase to a particular location. Such a rule could instruct all gem data objects containing the phrase "trademark" to be placed into the "Patent and Trademark Office" container. In another example, if the gem data object comes from a source, such as a URL, that is known to provide offensive material, the data could be discarded instead of being stored.

The pre-set rules on the server module may also relate to adding or modifying certain metadata associated with gem data objects. Such rules would allow a user to perform large amounts of data gathering without having to interact with each individual data set as it is collected. For example, a user could create a rule to apply a default bibliography to a gem data object for a certain source author. Some examples of pre-set server rules are:

If <dataobject> contains <keywords: x, y, x> then <locate in> <container C>

If <dataobject.author> contains <text: x, y, z> then <copy bibliography> <from item F>

After the application of any pre-set rules and routing has occurred, the gem data object is stored in step 1140. As noted above, this may be accomplished using a variety of different types of persistent storage 80.

D. Performing Data Actions Phase

There are a variety of data actions that may be performed using the electronic data-gathering system. For example, individual gem data objects may be retrieved, modified, deleted, or reorganized. Individual gem data objects may also be copied or cross-referenced to other locations within the database. A cross-reference is a pointer to a gem data object that is located elsewhere within the database. The gem object database may be searched, sorted, or reorganized. The contents of the gem object database may be displayed, for example in a hierarchical format showing the location of gem data objects within different containers.

Fig. 12 is a flowchart of a performing data actions phase for a research workflow method. In one embodiment, these actions involve interactions between the client module 40 and the server module 70. The client module 40 is responsible for identifying data sets, performing a first-pass analysis of the data, presenting a user interface to the user, and requesting single gem data objects or groups of objects according to search and browse interactions with the user. The server module 70 responds to search and browse requests, modifies data, processes detailed data analysis, and performs automated data routing where applicable.

For example, if a system user drags a gem data object from the data viewer 62 to a word processing program, the client module 40 generates a request to the server module 70 to retrieve the requested gem data object. The server module 70 responds by providing the gem data object or returning an error message. The client module 40, via the system clipboard data manager 60, takes the gem data object and evaluates its data. The client module 40 then makes the data available to the word processing application via the system clipboard.

Referring to Fig. 12, in step 1210, a request is received from the client module 40. The request is decoded in step 1220 so that it can be analyzed and processed.

The request is checked to see if the instructions are to add a new gem data object to the gem object database (step 1230). If yes, the request is then checked to determine if a filing preference has been included to route the new gem data object within the gem object database

(step 1232). If no filing preference is included, the server module 70 checks to see if there are pre-set rules which determine the routing of the gem data object (step 1236). The gem data object is then routed according to the included filing preferences or pre-set rules (step 1234).

However, if no filing preference or pre-set rules exist, the new gem data object will be routed to the "Unfiled" category or gem data object container (step 1238).

If the request is not to add a new gem data object, the request is checked to see if a data update is requested (step 1240). If yes, the server module 70 processes the request to update data by either editing or deleting data (step 1242). Examples of a data update request include modifying an existing gem data object, or cross-referencing or copying a gem data object to another location within the gem object database.

Otherwise, the request is an information-only query, which is processed by the server module 70 as shown in step 1250. Examples of an information-only query include expanding a node in a hierarchical depiction of the gem object database to view the contents of a particular gem data object container, or requesting a report on the database contents. A user could request a report based upon a search of the gem object database for a particular keyword or gem data object field. Some examples are:

If <dataobject> contains <keywords: x, y, x> then <display listing of dataobject>

If <dataobject.author> contains <text: x, y, z> then <display dataobject>

Upon completion of the request, the server module 70 encodes the response in step 1260 and returns the response to the client module 40 in step 1270. As discussed regarding the storage of gem data objects, a variety of different implementations may be used to encode and process client module 40 requests. One embodiment is an XML-RPC method call, but other embodiments will be evident to one of skill in the art.

Additional tasks may also be performed within the research system. As discussed above, the server module 70 responds to requests for information from the client module 40 by querying the gem object database. Additionally, the server module 70 may perform background tasks to aid the user of the electronic data-gathering system in identifying other potential sources of research information. A data analysis process is performed by the data analysis processor 78 as a background server module 70 task. The purpose of the data analysis process is to analyze the user of the electronic data-gathering system, as well as the user's

topics of interest, to make suggestions for further data gathering which may be forwarded to the client module 40.

The data analysis process performs two types of analysis passes. First, a general profile of the user of the electronic data-gathering system is created. Second, specific profiles are created for one or more topics of interest. If multiple users exist for the gem object database, the data analysis process will create individual user-specific profiles, and may additionally create profiles for topics of interest that have multiple users.

Creating a user profile is performed by deriving generalizations about the type of data stored by the user. Such an analysis may include, but is not limited to: the naming and depth of storage containers and sub-containers; the keywords associated with gem data objects; and the sources for the gem data objects. The data so obtained may then be compared against internally maintained demographics tables to determine the system user's most likely occupation, level of computer sophistication, areas of interest, most valued information sources, and various other attributes. The data may also be compared against databases provided by external sources. The data analysis process uses a similar technique to focus in on a specific topic of interest within the gem object database.

In one embodiment, the data analysis process compares the text of the gem data objects against stored tables of word roots. This process is most effective for areas of expertise that are well defined. For example, medical, legal, and technology professionals share vocabularies that can be distilled into word roots. The presence of these roots within the captured data can reveal primary and secondary occupations or areas of interest.

The data analysis process then uses the user profile and topical profiles to generate additional potential areas of research for the system user. Queries based upon the profiles are submitted to Internet search engines to generate Web site addresses of potential interest. Queries are also submitted to the query processor 76 of the server module 70, which queries the local gem object database and any other local files systems for additional gem data objects or files of potential interest. The queries may also suggest other users of the system with similar interests or research topics, thereby fostering collaboration between users who may not know each other.

For example, the data analysis process may reveal that a particular user has created a gem data object collection with the following hierarchy of containers:

The U.S. Constitution
- Articles
- Amendments
- Recent decisions

Legal Research
- Patents
  - The U.S. PTO

Personal
- DVD Players
- Vacations
  - Bali
  - Jamaica
  - Hawaii

Given this example, the data analysis process would infer that: the system user is a lawyer or legal scholar who is currently interested in Constitutional and Patent law; the user has a medium to high computer sophistication; the user may be shopping for a new DVD player; and the user may take a vacation to a tropical island in the near future. Based upon this information, the data analysis process would submit a query such as "tropical island vacation" to one or more Internet search engines and return the top ten resulting list of sites as a suggestion to the system user for future research. The data analysis processor 78 would also submit a query to the local query processor 76 to check local files and gem object databases for information regarding tropical island vacations. The data analysis process would perform similar queries for the other topical areas of research found.

One embodiment of a data analysis process is show in Fig. 13. In step 1300, a pass through all data containers is made to search for keywords and phrases. Text labels are parsed, and small words are discarded. In step 1310, a data pass is made through all individual gem data objects to search for keywords and phrases. The data is parsed and small words are again discarded. In step 1320, a count is made of the most used data sources, such as URLs, local files, and network or system files. Steps 1300-1320 may be performed in any order.

In step 1340, a comparison is made between the information obtained in steps 1300-1320 and a stored database of word roots 1330. The stored database of word roots provides an association between common data found in steps 1300-1320 and particular professions or

fields of study. For example, the root "ombro" is shown in the database of word roots 1330 as being associated with medicine and psychology.

The comparison of step 1340 generates additional information. A user profile 1350 is generated, containing: an occupation and/or topics of research; favored data sources; and suggested search queries. In step 1360, these suggested types of search queries are submitted and the results evaluated. In step 1370, additional gem data objects are created out of the results of the search query. Finally, in step 1380 the newly-created gem data objects are placed in a container within the user's gem object database.

E. Examples of Gem Data Objects

Figs. 14-17 show one embodiment of a gem data object formatting scheme. This gem data object scheme may be used to implement a client module and a server module that operate upon the gem data objects and allow users to interact with a gem object database. It should be understood that the coding and formatting scheme for implementing the electronic data-gathering system is not limited to the embodiments disclosed herein. Other formatting methods will be evident to one of skill in the art.

The gem data objects in the scheme shown in Figs. 14-17 are organized in a hierarchical "tree" structure. Within the tree structure, there is a "root" node that contains the tree structure within the gem object database. All members of the tree structure are part of this root node. There are also various levels of container objects that represent inner nodes and empty nodes of the tree structure. Finally, the "leaf" nodes of the tree structure are represented by individual items, corresponding to individual gem data objects. The example structure of Figs. 14-17 is directed towards a data tree that contains information about scary books.

Fig. 14 shows an example of a root node. A root node may contain only other containers. The root created in Fig. 14 is named "Examples", and becomes the root node of a tree containing examples of scary books. As given in line 1410, the server Universal Resource Identifier (URI) for this root node is "gemserver.gemteq.com." This URI allows the client and server modules to locate the root node.

Fig. 15 shows an example of a container node. A container node may contain other containers or individual gem data objects, or both. The container of Fig. 15 is named "ScaryBooks." As given in line 1509, the "ScaryBooks" container has a URI of

"gemserver.gemteq.com/Examples." This URI indicates the "ScaryBooks" container is located in the "Examples" root node.

Fig. 16 shows another example of a container. The container of Fig. 16 is named "StephenKing." This container has a URI of "gemserver.gemteq.com/Examples/ScaryBooks", as given in line 1609. This URI indicates the "StephenKing" container is located in the "ScaryBooks" container in the "Examples" root node.

Fig. 17 shows an example of an individual item. The item of Fig. 17 is named "Carrie". As given in line 1709, this item has a URI of "gemserver.gemteq.com/Examples/ScaryBooks/StephenKing." This URI indicates the "Carrie item is located in the "StephenKing" container, in the "ScaryBooks" container, in the "Examples" root node.

A client would use the above example of a tree structure to find and route gem data objects. For example, the client module 40 would parse the URI of the "Carrie" item shown in Fig. 17 in the following way. The server module 70 would be indicated as "gemserver.gemteq.com". The data tree would be given by the root node "Examples". The path would be given by the two container nodes "ScaryBooks/StephenKing". The actual gem data object would be the item "Carrie."

In an XML-based embodiment, using this path information, the client module 40 would make a RPC to the server module 70 using standard HTTP protocols to request the "Carrie" gem data object. The server module 70 would then reply with a standard HTTP response that contains the XML-RPC response that in turn contains the "Carrie" gem data object requested. The XML portion of the response may be of many different types to represent the success or failure of the RPC call.

The embodiment of a gem data object "Carrie" as shown in Fig. 17 demonstrates that a gem data object is an encapsulation of the original electronic data combined with creation, attribution, and user-selected metadata. Line 1703 of the "Carrie" gem data object contains user-selected comments about the gem data object. Lines 1704-1710 of the "Carrie" gem data object contain metadata about the creation of the original gem data object: when the gem data object was created (line 1704); who created the gem data object (lines 1705-1708); and when the gem data object was modified (line 1710). Lines 1711-1722 contain attribution metadata as well as information about how a bibliography for the "Carrie" gem data object should be

formatted. For example, line 1711 gives the bibliography format "MLABook", which corresponds to one of the bibliography format choices from the popup menu 924 of Fig. 9B.

The original data for the gem data object "Carrie" is also shown in Fig. 17. Lines 1729-1731 list the text data that is encapsulated with the metadata listed above. The example "Carrie" is a text type of gem data object. However, a variety of other types of gem data objects are also available.

Several different types of gem data objects are shown in Figs. 18-22. The overall structure for a gem data object will remain substantially the same regardless of the type of original data contained in the gem data object. Examples of data types that may comprise the original data have been discussed previously with regard to Figs. 4A and 4B. Each type of gem data object will contain the original data encapsulated with creation, attribution, and user-selected metadata. This allows for consistency in the use of gem data objects, and also allows new types of gem data objects to be easily incorporated into the gem object database. The gem data object types shown in Figs. 18-22 are merely examples of specific embodiments. It will be evident to one of skill in the art that various other types of gem data objects may be created, stored, and manipulated within the electronic data-gathering system.

Figs. 18-22 show a user interface window for a gem data object viewer. In each figure, the "View" tab of the popup menu for a different gem data object is expanded. This creates a view of the original data encapsulated within the gem data object.

Fig. 18 shows in a window 1870 a view of the gem data object "Functions of Patent and Trademark Office" which is a text type of gem data object. Fig. 19 shows in a window 1970 another text type of gem data object "Text of plant patent form." As shown in the window 1970, text type gem data objects may contain different text formatting styles as well as plain text. Both ASCII and RTF text formats are supported, as well as a variety of word processing applications' native formats.

Fig. 20 shows in a window 2070 a view of a file type of gem data object. The file gem data object "My briefcase - link" provides a link to the file "plantpat.doc." As shown by this file gem data object, it is possible to mark a set of data for research applications not only by saving the data itself, but by saving a link to the path and filename of the data which can later be used to retrieve the data.

A related type of gem data object is an OLE object. A OLE type gem data object contains: the data required to present the source data embedded in a target document; data about the providing application; and either a path to the data file or the raw data itself.

Fig. 21 shows in a window 2170 a pictorial type of gem data object. The window 2170 contains a view of the picture file "PTO header picture". Graphical file types, such as JPEG and GIF formatted files, may be stored as pictorial gem data objects

Fig. 22 shows in a window 2270 a view of a URL type of gem data object. The URL gem data object "Patent and Trademark Office Home Page" contains a URL that provides an Internet link to the home page of the Patent and Trademark Office on the World Wide Web (WWW). Such URL type gem data objects may be created where the user does not wish to store the entire web page as a separate gem data object.

File types may be stored in clipboard format or converted from native formats. Standard clipboard formats are available for a variety of data types, including: plain text (ASCII); rich text format; HTML text; Bitmap Image (BMP); Device Independent Bitmap Image (DIB); Metafile Image (WMF); Enhanced Metafile Image (EMF); and Icon Image (ICO). Other formats, such as GIF or JPEG, may be converted by the system clipboard or the electronic data-gathering system.

Although the invention has been described in considerable detail with reference to certain embodiments, other embodiments are possible. As will be understood by those of skill in the art, the invention may be embodied in other specific forms without departing from the essential characteristics thereof. For example, different formats may be used to create gem data objects. Additionally, different types of client/server architecture may be used to implement the electronic data-gathering system. Accordingly, the present invention is intended to embrace all such alternatives, modifications and variations as fall within the spirit and scope of the appended claims and equivalents.

We claim:

1. In a computer system including a user interface and a persistent storage mechanism, a computer implemented method for encapsulating user-selected sets of electronic data, the method comprising:

acquiring a set of data; and

populating a plurality of gem data object field(s) with the set of data encapsulated with attribution, creation, and user-defined metadata to create a gem data object.

2. The method of claim 1 further comprising:

packaging the set of data into an interim object by appending a set of attribution, creation, and user-defined metadata before creating a gem data object.

3. The method of claim 1 further comprising:

storing a gem data object format field representing a proper format for an attribution caption for the gem data object; and

formatting attribution and source metadata based upon the format field.

4. The method of claim 3 wherein the attribution caption is a bibliography.

5. In a computer system including a user interface and a persistent storage mechanism, a computer implemented method for encapsulating a user-selected set of electronic data, the method comprising:

receiving a definition of the set of data in electronic format;

capturing the set of data without opening a new program view within the user interface; and

populating a gem data object field with the captured set of data to create a gem data object.

6. The method of claim 5 including:

packaging the captured set of data into an interim object by appending a set of creation and attribution metadata before populating the gem data object field.

7. The method of claim 6 wherein the set of metadata includes computer session information.

8. The method of claim 6 wherein the set of metadata includes source information regarding the set of data.

9. The method of claim 6 wherein the set of metadata includes bibliographic information regarding the set of data.

10. The method of claim 6 wherein the set of metadata includes a default name and description of the set of data.

11. The method of claim 6 wherein the set of metadata includes a set of keywords in the set of data.

12. The method of claim 6 wherein the set of metadata includes a Uniform Resource Locator from the set of data.

13. The method of claim 5 including:

storing the gem data object in a persistent storage mechanism.

14. The method of claim 5 wherein the set of data in electronic format is defined by a user interacting with the system clipboard.

15. The method of claim 5 wherein the set of data in electronic format is defined by a user performing drag and drop activity.

16. The method of claim 5 wherein the set of data in electronic format is defined by a user manually typing data into a gem data object user interface.

17. The method of claim 5 wherein the set of data in electronic format is defined by a user using a scanner.

18. The method of claim 5 wherein the gem data object is handled by a server module which interacts with a persistent data storage mechanism.

19. The method of claim 18 wherein the server module is independent of the persistent data storage mechanism implementation or vendor.

20. The method of claim 18 wherein access to the persistent data storage mechanism is controlled by a security system consisting of user and group accounts, and permissions.

21. The method of claim 18 wherein the server module is inprocess or remote from a client module.

22. The method of claim 21 wherein the server module communicates with the client module via in-process communication mechanisms.

23. The method of claim 21 wherein the server module communicates with the client module via standard Internet communication mechanisms.

24. The method of claim 18 wherein the server module applies definable routing rules to the gem data object.

25. The method of claim 24 wherein the definable routing rules allow the server module to determine the location in a database in which the gem data object will be stored.

26. The method of claim 24 wherein the definable routing rules instruct the server module to discard the gem data object if the gem data object contains pre-specified data.

27. The method of claim 18 wherein the server module includes a data analysis process operating on the data stored in the persistent data storage mechanism.

28. The method of claim 27 wherein the data analysis process may determine a profile of a user of either the gem data object or the computer system.

29. The method of claim 28 wherein the profile identifies the occupation of the user.

30. The method of claim 28 wherein the profile identifies a topic of interest of the user.

31. The method of claim 28 wherein the profile is used to suggest a research topic for a first user of either the gem data object or the computer system.

32. The method of claim 31 wherein the research topic is an Internet site.

33. The method of claim 31 wherein the research topic is a local system file.

34. The method of claim 31 wherein the research topic is a gem data object.

35. The method of claim 31 wherein the research topic identifies a second user of either the gem data object or the computer system.

36. The method of claim 27 wherein the data analysis process may determine the nature or subject of data-gathering done by a user of either a first gem data object or the computer system.

37. The method of claim 36 wherein the data analysis process suggests a further research topic to the user.

38. The method of claim 36 wherein the data analysis process suggests an Internet site to the user.

39. The method of claim 36 wherein the data analysis process suggests a local file to the user.

40. The method of claim 36 wherein the data analysis process suggests a second gem data object to the user.

41. A method for using one or more gem data object(s) in research applications, comprising:

accessing a stored set of one or more gem data objects, wherein the one or more stored gem data object(s) include:

a set of encapsulated data; and

a set of creation and attribution metadata; and

manipulating one or more of the gem data object(s) within the stored set.

42. A computer program product for an electronic data-gathering system, comprising:

a computer readable medium that stores program code including:

program code that acquires a set of data; and

program code that populates a plurality of gem data object field(s) with the set of data encapsulated with attribution, creation, and user-defined metadata to create a gem data object.

43. The computer program product of claim 42, further comprising:

program code that packages the set of data into an interim object by appending a set of attribution, creation, and user-defined metadata before creating a gem data object.

44. The computer program product of claim 42, further comprising:

program code that stores a gem data object format field representing a proper format for an attribution caption for the gem data object; and

program code that formats attribution and source metadata based upon the format field.

45. A computer program product for an electronic data-gathering system, comprising:

a computer readable medium that stores program code including:

program code that receives a definition of the set of data in electronic format;

program code that captures the set of data without opening a new program view within the user interface; and

program code that populates a gem data object field with the captured set of data to create a gem data object.

46. The computer program product of claim 45, further comprising:

program code that packages the captured set of data into an interim object by appending a set of creation and attribution metadata before populating the gem data object field.

47. The computer program product of claim 45, wherein the gem data object is handled by a server module which interacts with a persistent data storage mechanism.

48. The computer program product of claim 47, wherein the server module applies definable routing rules to the gem data object.

49. The computer program product of claim 47, wherein the server module includes a data analysis process operating on the data stored in the persistent data storage mechanism.

50. The computer program product of claim 49, wherein the data analysis process may determine a profile of a user of either the gem data object or the computer system.

51. The computer program product of claim 49, wherein the data analysis process may determine the nature or subject of data-gathering done by a user of either a first gem data object or the computer system.

52. A computer readable memory for storing original data packaged with attribution and creation metadata for access by an application program being executed on a data processing system, comprising:

a data structure stored in said memory, the data structure including information resident in a database used by the application program and including:

an encapsulation of the original data;

a set of user-specific metadata; and

a set of attribution and creation metadata, wherein the set of attribution and creation metadata is automatically added to the original data.

53. The computer readable memory of claim 52 wherein the original data is comprised of text data.

54. The computer readable memory of claim 52 wherein the original data is comprised of pictorial data.

55. The computer readable memory of claim 52 wherein the original data is comprised of an embedded object formatted in the Object Linking and Embedding standard.

56. The computer readable memory of claim 52 wherein the original data is comprised of a Universal Resource Locator address.

57. The computer readable memory of claim 52 wherein the original data is comprised of text in Hyper Text Markup Language format.

58. The computer readable memory of claim 52 wherein the original data is comprised of the path and filename of the original source.

59. The computer readable memory of claim 52 wherein the data structure contains programming code allowing an attribution caption to be automatically created and formatted for the data structure.

60. The computer readable memory of claim 59 wherein the attribution caption is a bibliography.

61. A computer data signal for storing original data packaged with metadata embodied in a transmission medium comprising:

an encapsulation of the original data;

a set of user specific metadata; and

a set of attribution and creation metadata, wherein the set of attribution and creation metadata is automatically added to the original data.

1/24

System Overview

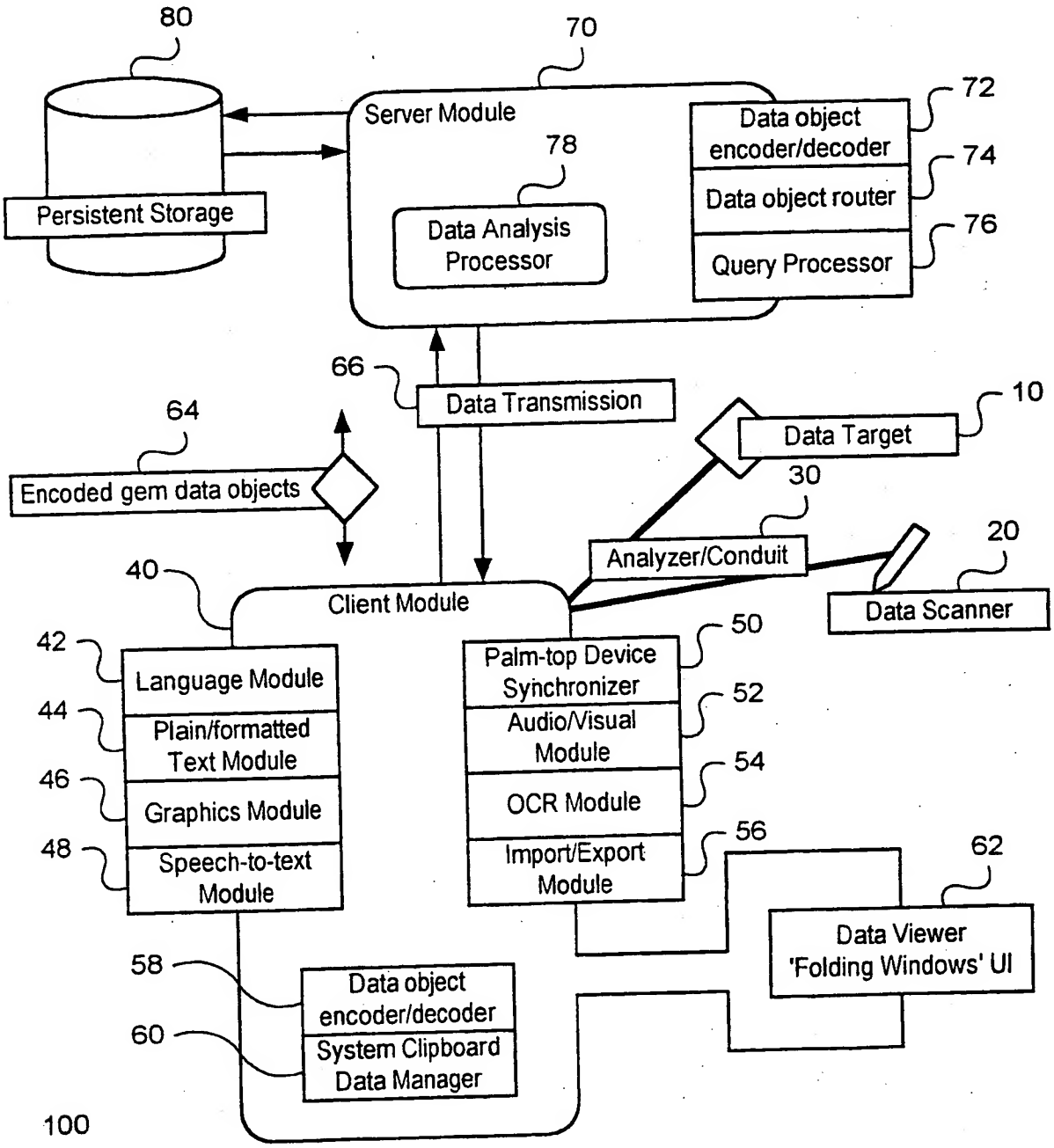


FIGURE 1

Research Workflow

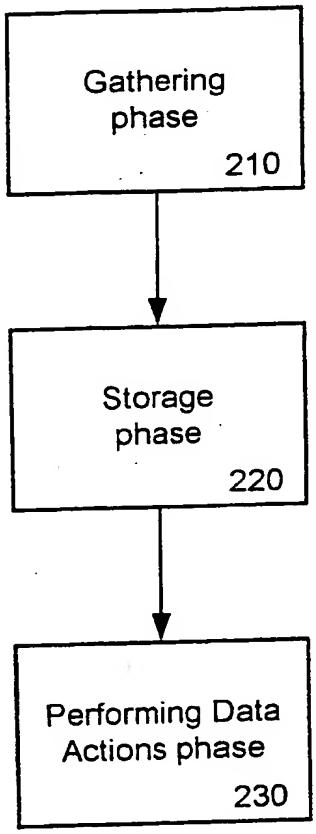


FIGURE 2

Gathering Phase

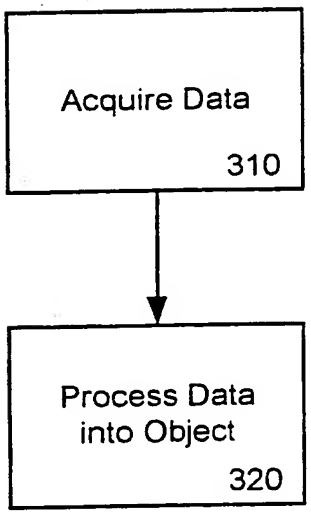


FIGURE 3

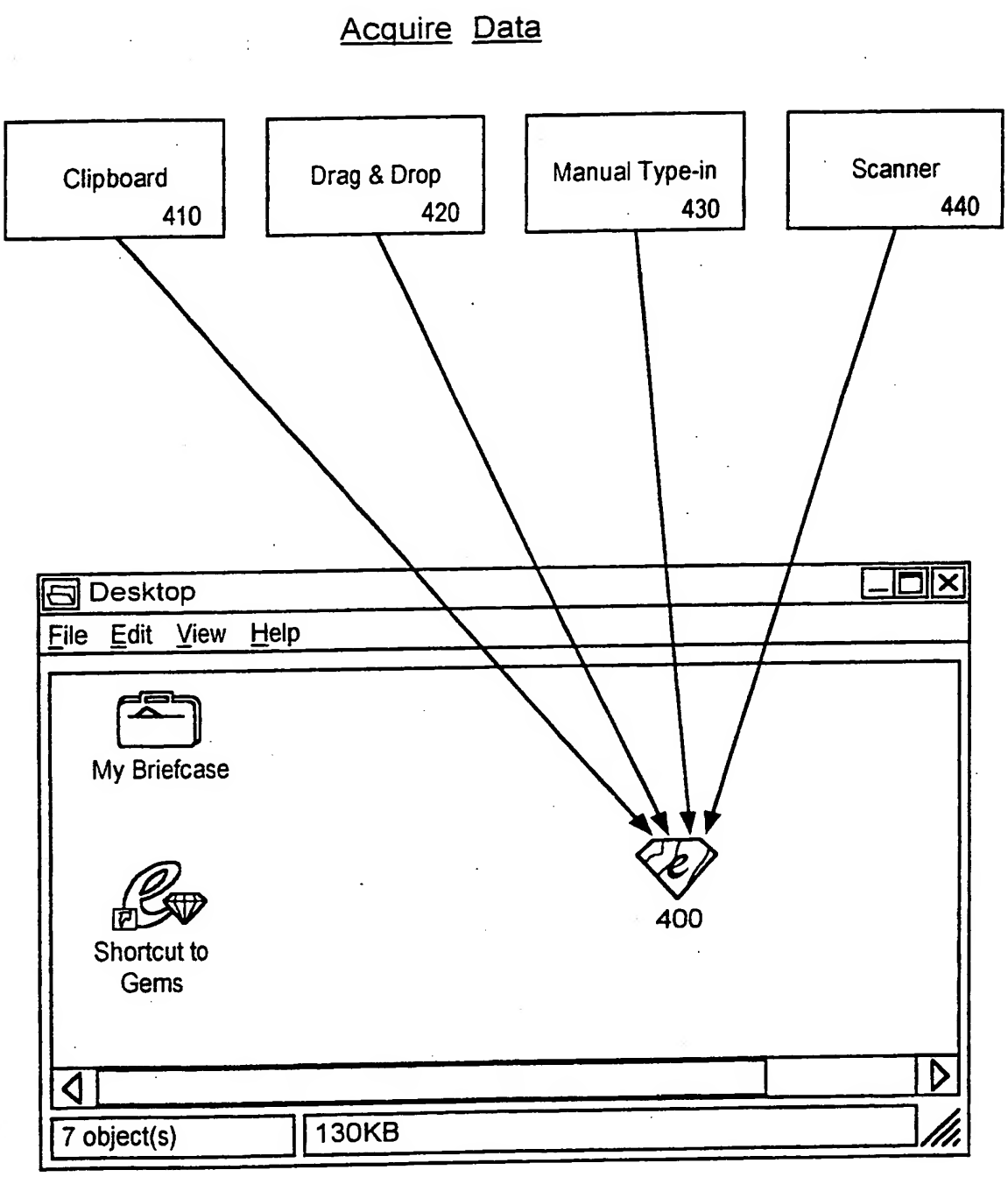

Acquire Data
Clipboard
410
Drag & Drop
420
Manual Type-in
430
Scanner
440
Desktop
File Edit View Help
My Briefcase
Shortcut to
Gems
400
7 object(s)
130KB

FIGURE 4A

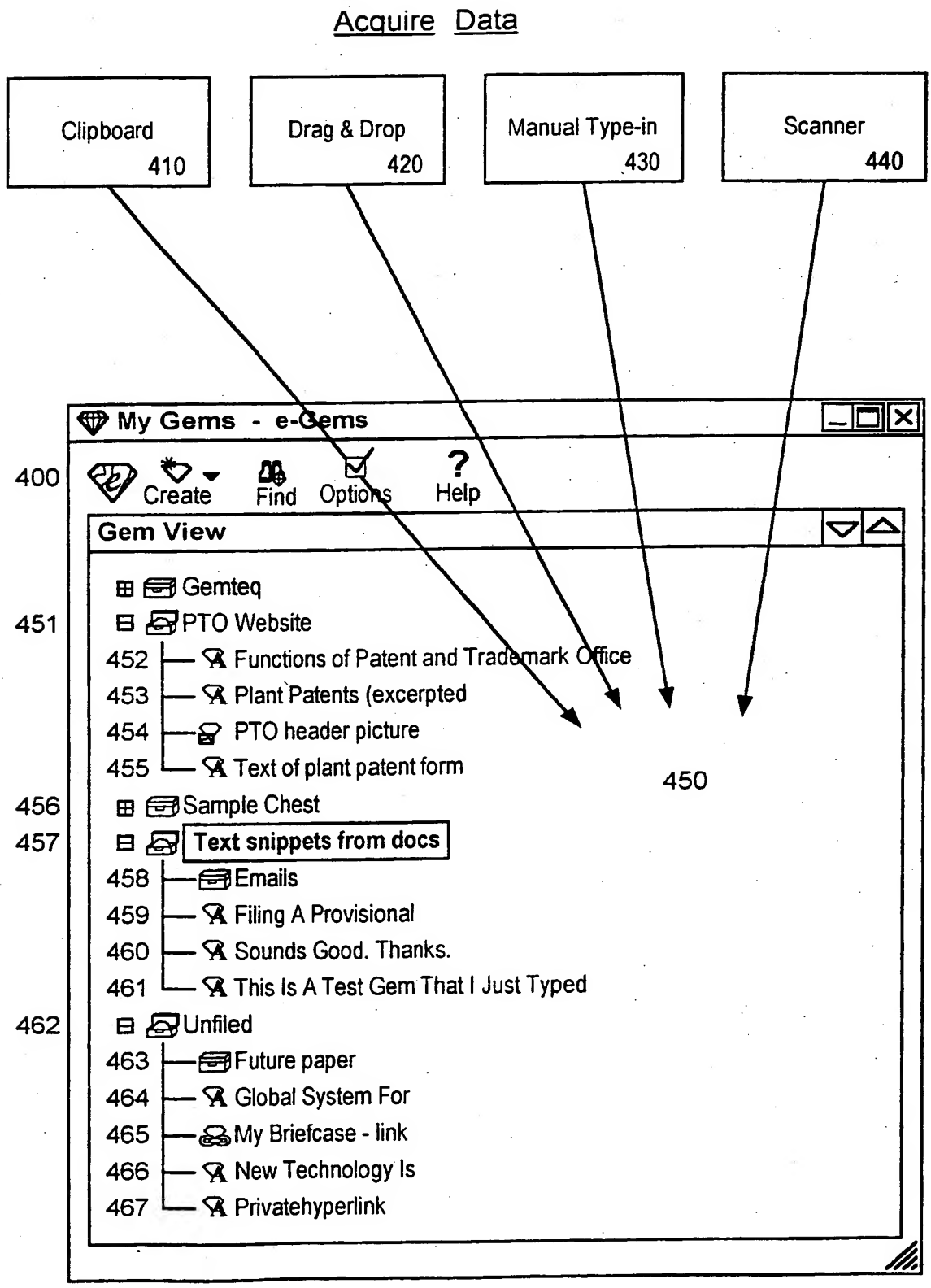

Acquire Data
Clipboard 410
Drag & Drop 420
Manual Type-in 430
Scanner 440
My Gems - e-Gems
400
Create
Find
Options
Help
Gem View
Gemteq
451 PTO Website
452 Functions of Patent and Trademark Office
453 Plant Patents (excerpted
454 PTO header picture
455 Text of plant patent form
450
456 Sample Chest
457 Text snippets from docs
458 Emails
459 Filing A Provisional
460 Sounds Good. Thanks.
461 This Is A Test Gem That I Just Typed
462 Unfiled
463 Future paper
464 Global System For
465 My Briefcase - link
466 New Technology Is
467 Privatehyperlink

FIGURE 4B

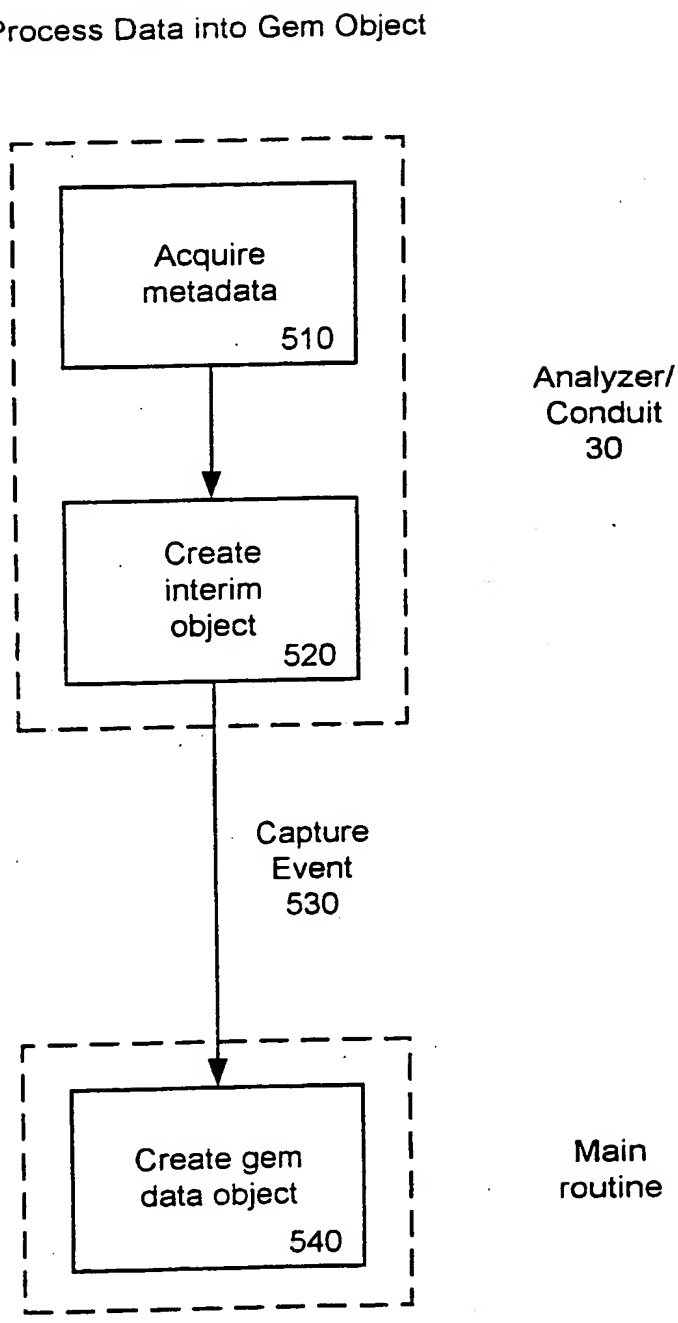

Process Data into Gem Object
Acquire metadata
510
Create interim object
520
Analyzer/ Conduit 30
Capture Event
530
Create gem data object
540
Main routine

FIGURE 5

Create Interim Object

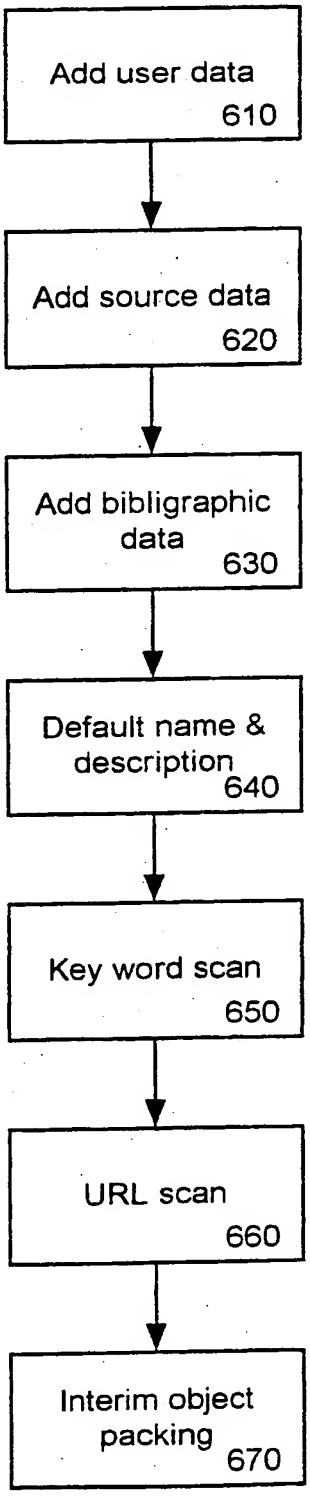


FIGURE 6

Create Gem Data Object

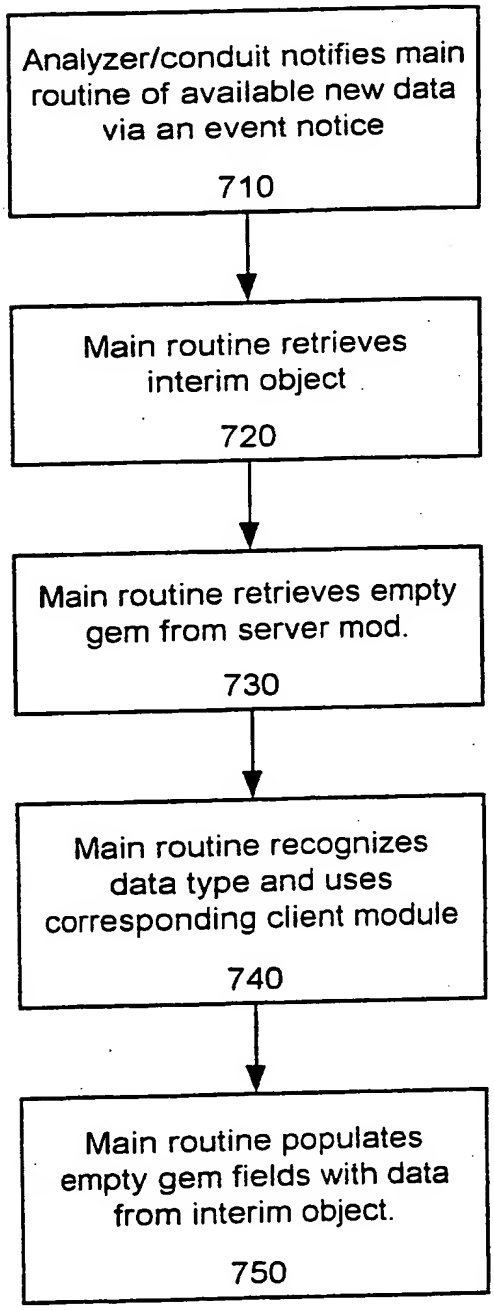


FIGURE 7

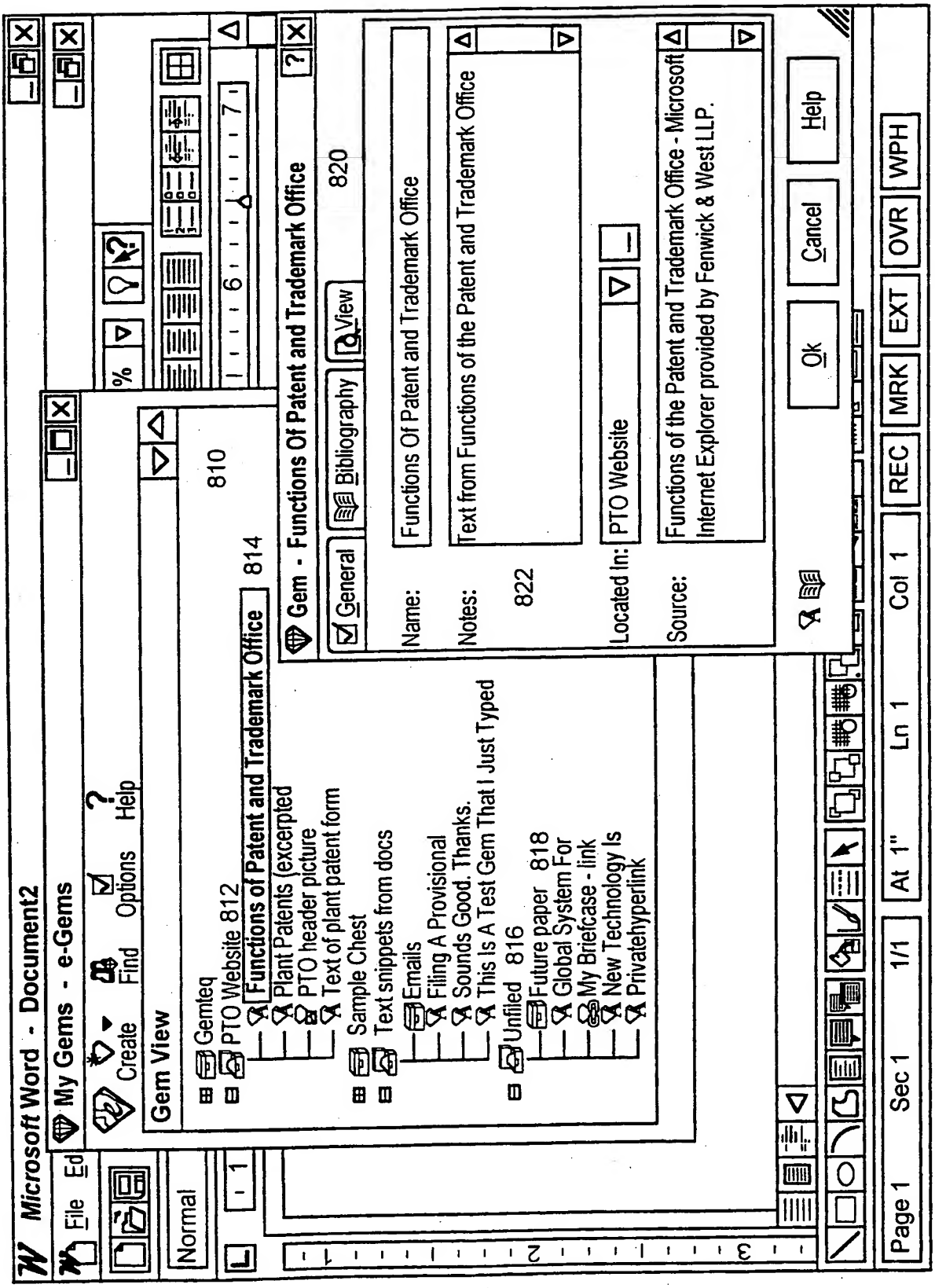

Microsoft Word - Document2
File Ed
My Gems - e-Gems
Create
Find
Options
Help
Normal
Gem View
Gemteq
PTO Website 812
Functions of Patent and Trademark Office 814
Plant Patents (excerpted
PTO header picture
Text of plant patent form
Sample Chest
Text snippets from docs
Emails
Filing A Provisional
Sounds Good. Thanks.
This Is A Test Gem That I Just Typed
Unfiled 816
Future paper 818
Global System For
My Briefcase - link
New Technology Is
Privatehyperlink
810
Gem - Functions Of Patent and Trademark Office
General
Bibliography
View
820
Name:
Functions Of Patent and Trademark Office
Notes:
Text from Functions of the Patent and Trademark Office
822
Located In:
PTO Website
Source:
Functions of the Patent and Trademark Office - Microsoft Internet Explorer provided by Fenwick & West LLP.
Ok
Cancel
Help
Page 1
Sec 1
1/1
At 1"
Ln 1
Col 1
REC
MRK
EXT
OVR
WPH

FIGURE 8

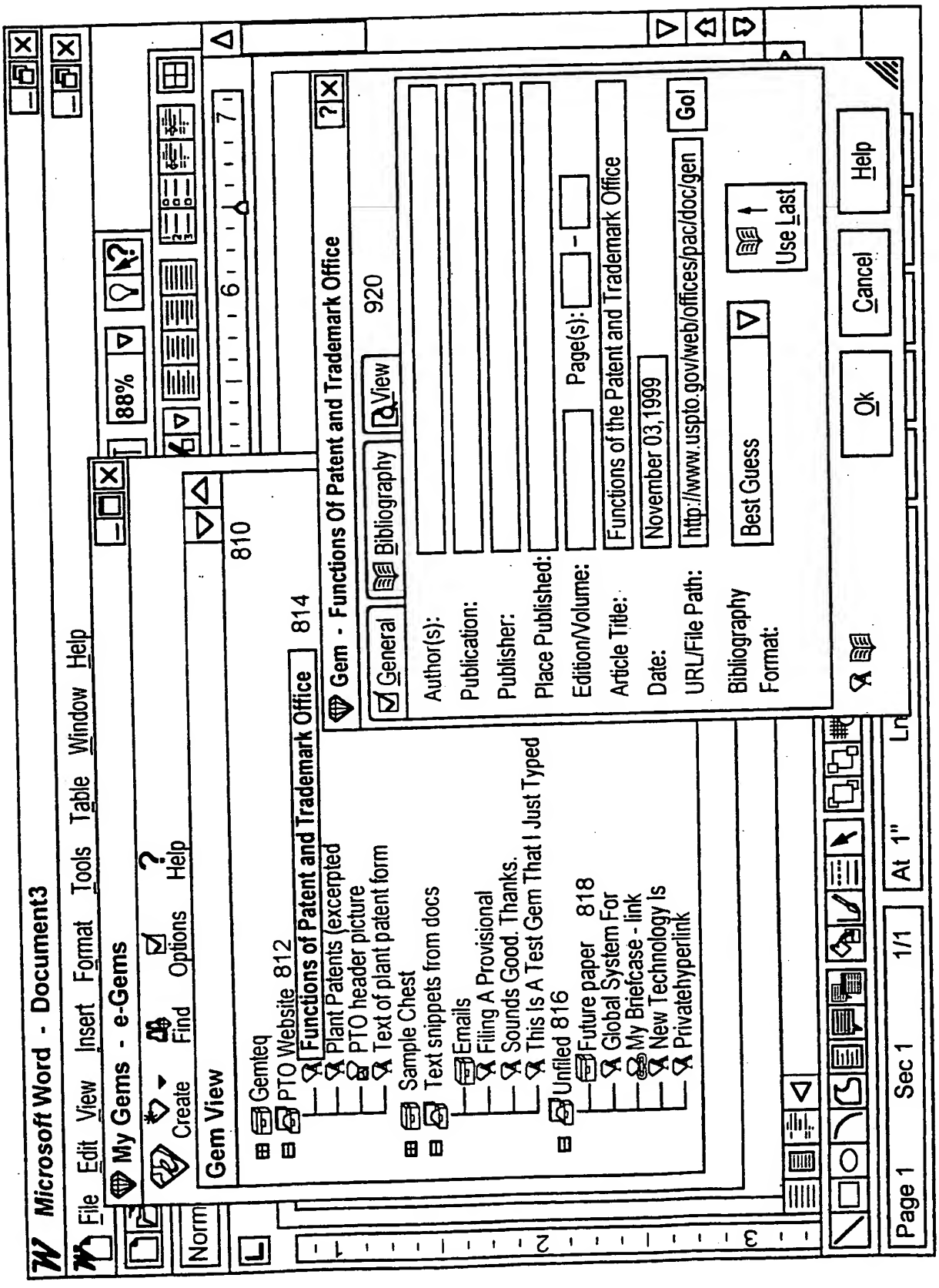

Microsoft Word - Document3
File Edit View Insert Format Tools Table Window Help
My Gems - e-Gems
Create Find Options Help
Gem View
810
Gemteq
PTO Website 812
Functions of Patent and Trademark Office 814
Plant Patents (excepted
PTO header picture
Text of plant patent form
Sample Chest
Text snippets from docs
Emails
Filing A Provisional
Sounds Good. Thanks.
This Is A Test Gem That I Just Typed
Unfiled 816
Future paper 818
Global System For
My Briefcase - link
New Technology Is
Privatehyperlink
Gem - Functions Of Patent and Trademark Office
General
Bibliography
View
920
Author(s):
Publication:
Publisher:
Place Published:
Edition/Volume:
Page(s):
Article Title:
Functions of the Patent and Trademark Office
Date:
November 03,1999
URL/File Path:
http://www.uspto.gov/web/offices/pac/doc/gen
Go!
Bibliography Format:
Best Guess
Use Last
Ok
Cancel
Help
88%
Page 1
Sec 1
1/1
At 1"

FIGURE 9A

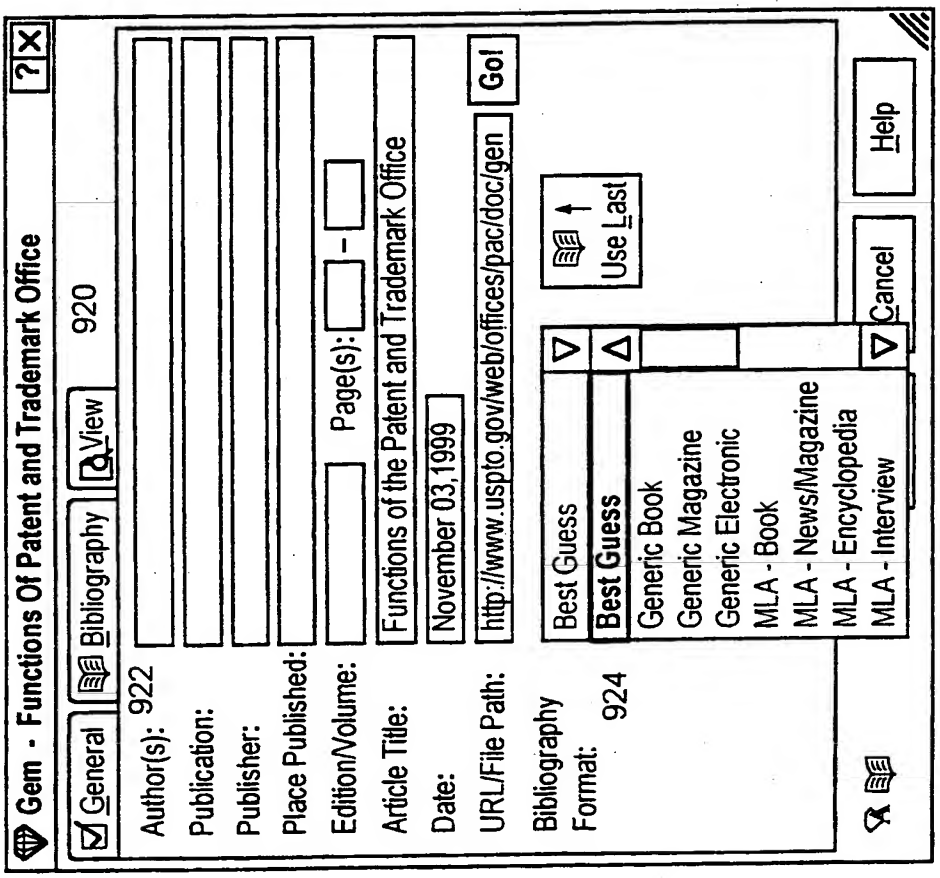

Gem - Functions Of Patent and Trademark Office
General
Bibliography
View
920
Author(s):
922
Publication:
Publisher:
Place Published:
Edition/Volume:
Page(s):
Article Title:
Functions of the Patent and Trademark Office
Date:
November 03,1999
URL/File Path:
http://www.uspto.gov/web/offices/pac/doc/gen
Go!
Bibliography Format:
924
Best Guess
Best Guess
Generic Book
Generic Magazine
Generic Electronic
MLA - Book
MLA - News/Magazine
MLA - Encyclopedia
MLA - Interview
Use Last
Cancel
Help

FIGURE 9B

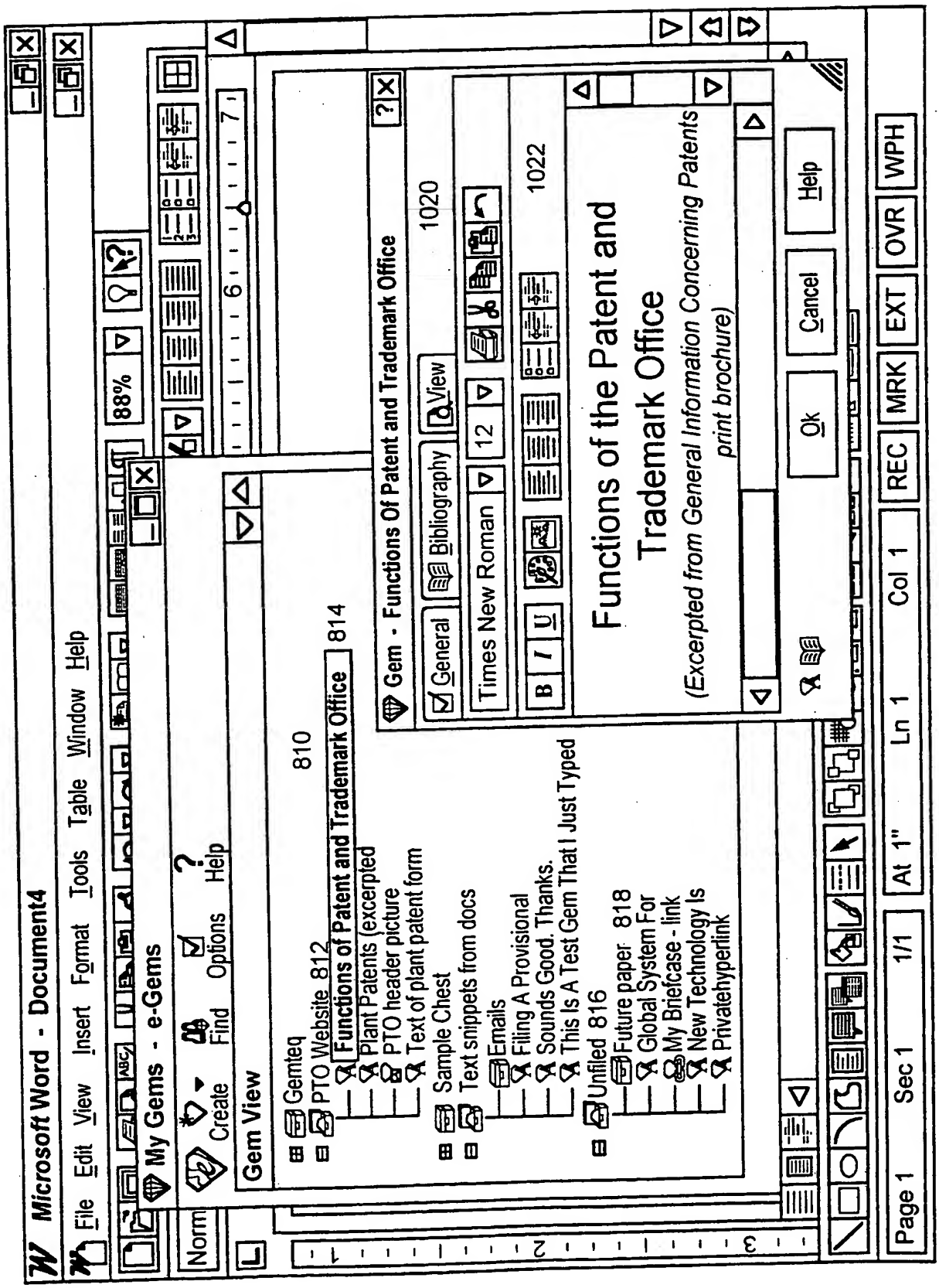

Microsoft Word - Document4
File Edit View Insert Format Tools Table Window Help
My Gems - e-Gems
Norm
Create Find Options Help
Gem View
Gemteq
PTO Website 812
Functions of Patent and Trademark Office 814
Plant Patents (excerpted
PTO header picture
Text of plant patent form
Sample Chest
Text snippets from docs
Emails
Filing A Provisional
Sounds Good. Thanks.
This Is A Test Gem That I Just Typed
Unfiled 816
Future paper 818
Global System For
My Briefcase - link
New Technology Is
Privatehyperlink
810
Gem - Functions Of Patent and Trademark Office
General
Bibliography
View
1020
Times New Roman
12
B I U
1022
Functions of the Patent and Trademark Office
(Excerpted from General Information Concerning Patents
print brochure)
Ok
Cancel
Help
Page 1 Sec 1 1/1 At 1" Ln 1 Col 1 REC MRK EXT OVR WPH

FIGURE 10

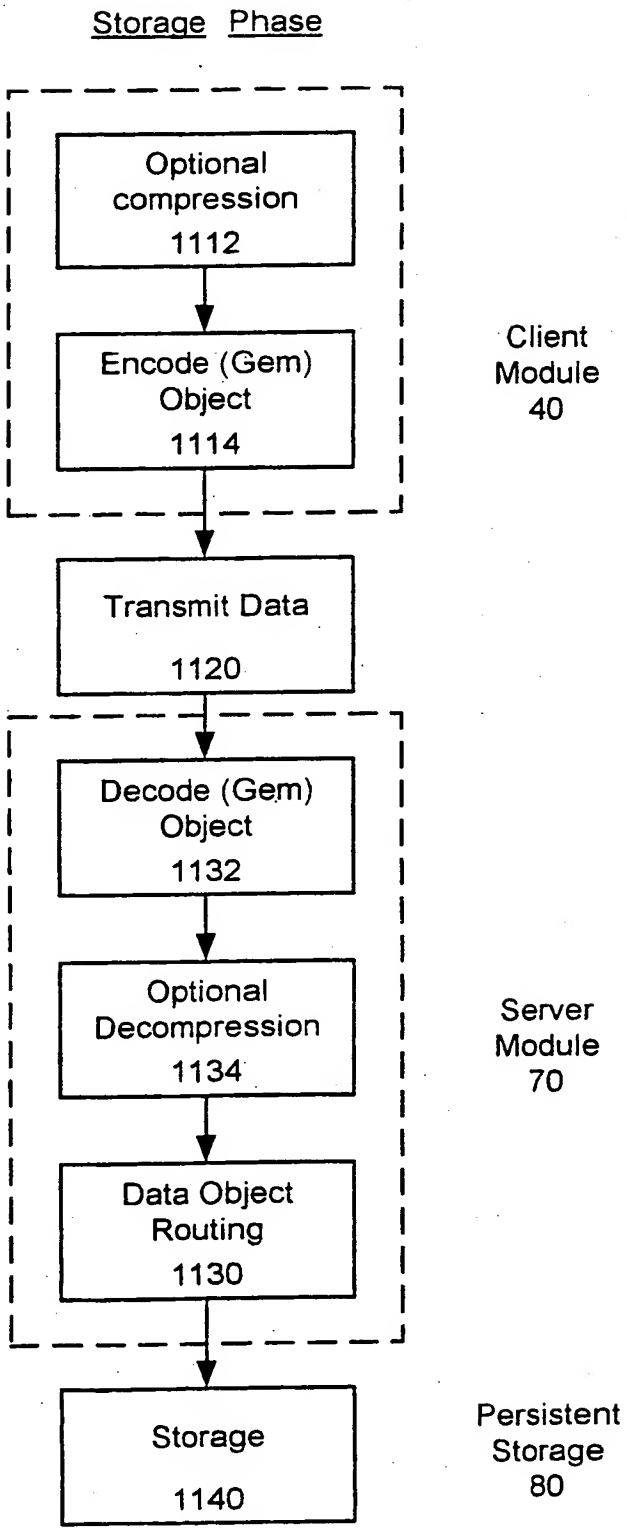

Storage Phase
Optional compression 1112
Encode (Gem) Object 1114
Client Module 40
Transmit Data 1120
Decode (Gem) Object 1132
Optional Decompression 1134
Server Module 70
Data Object Routing 1130
Storage 1140
Persistent Storage 80

FIGURE 11

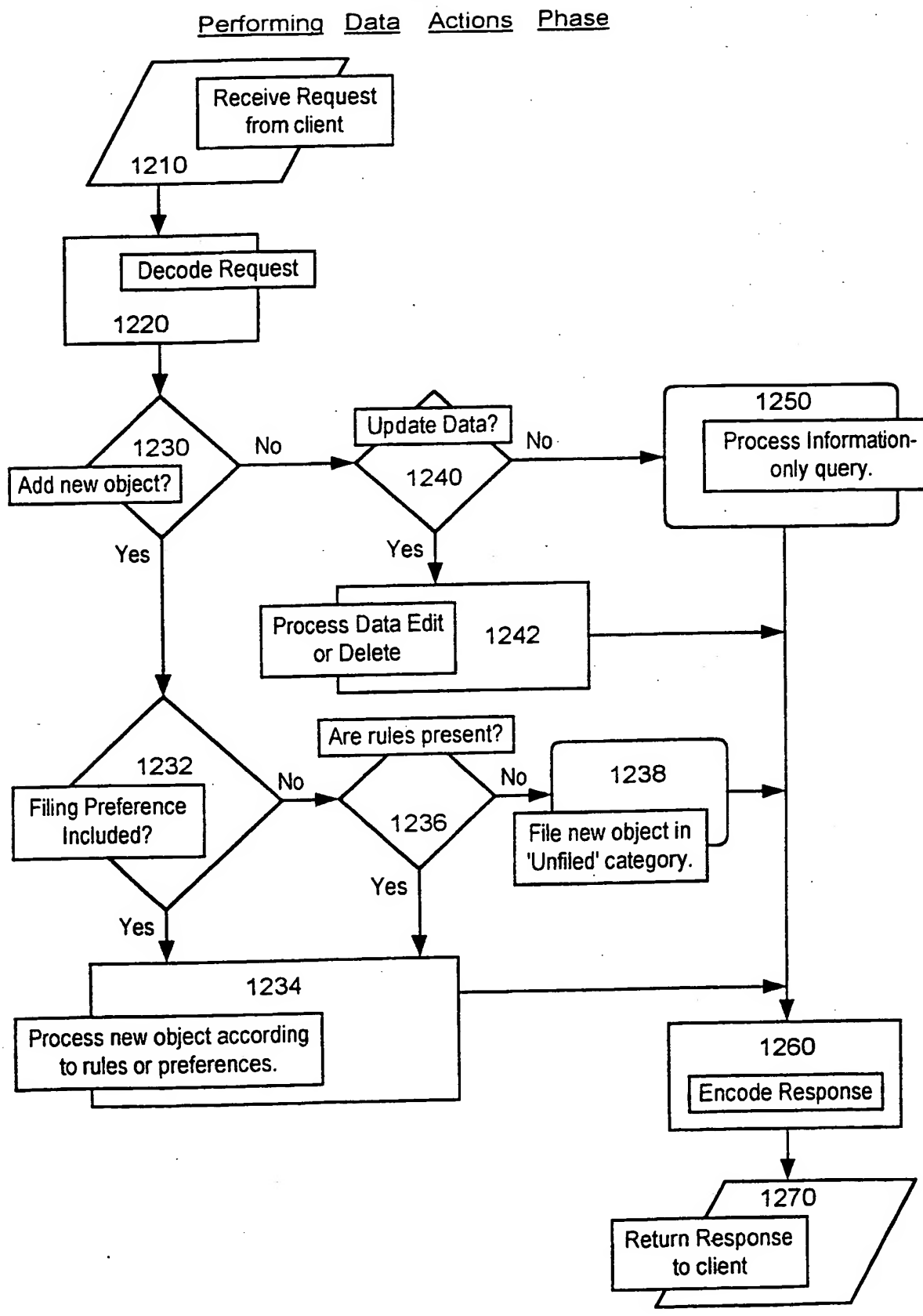

Performing Data Actions Phase
Receive Request from client
1210
Decode Request
1220
1230
Add new object?
No
Update Data?
1240
No
1250
Process Information-only query.
Yes
Yes
Process Data Edit or Delete
1242
1232
Filing Preference Included?
No
Are rules present?
1236
No
1238
File new object in 'Unfiled' category.
Yes
Yes
1234
Process new object according to rules or preferences.
1260
Encode Response
1270
Return Response to client

FIGURE 12

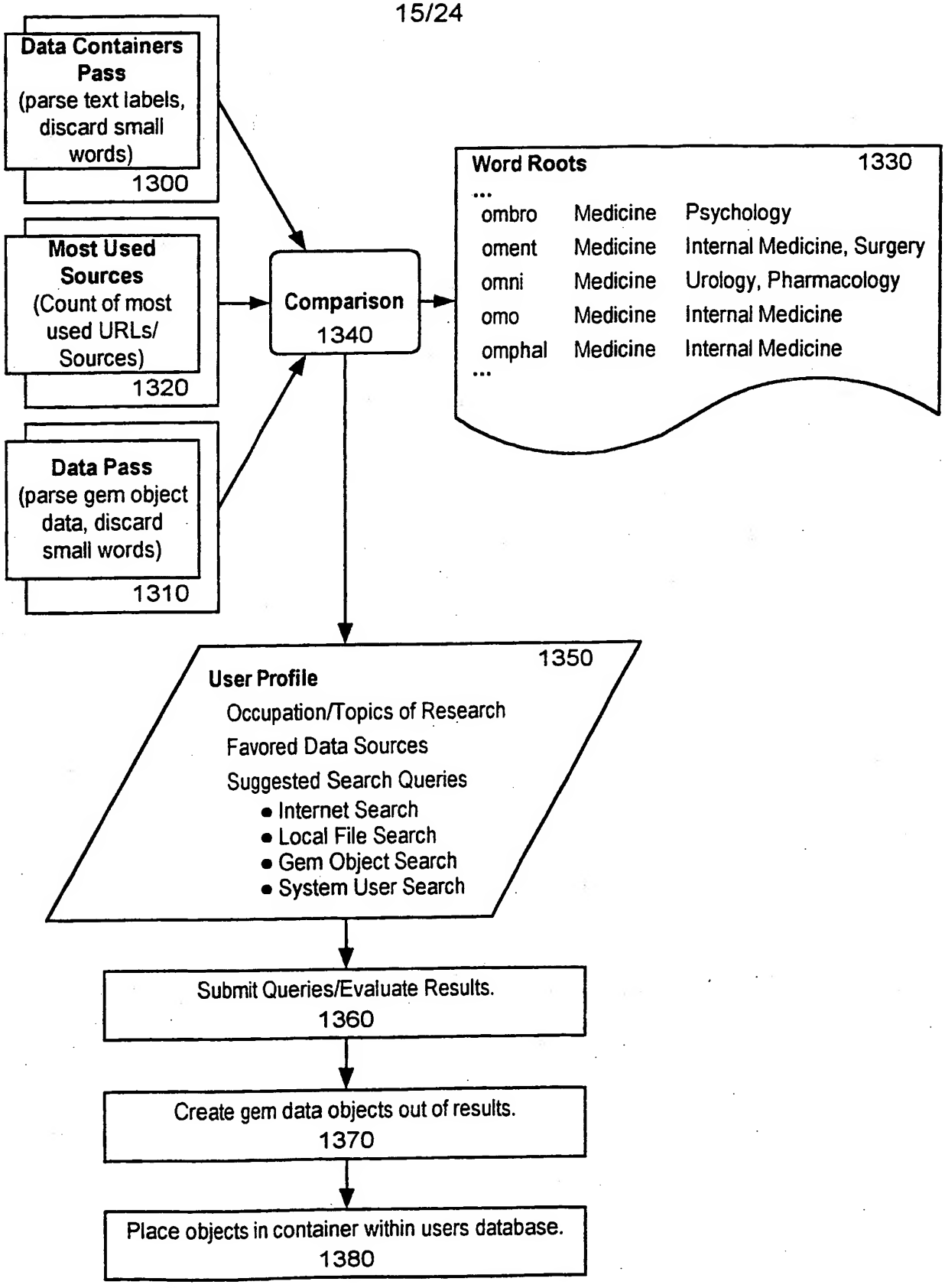

Data Containers Pass
(parse text labels, discard small words)
1300
Most Used Sources
(Count of most used URLs/ Sources)
1320
Data Pass
(parse gem object data, discard small words)
1310
Comparison
1340
Word Roots
1330
...
ombro Medicine Psychology
oment Medicine Internal Medicine, Surgery
omni Medicine Urology, Pharmacology
omo Medicine Internal Medicine
omphal Medicine Internal Medicine
...
1350
User Profile
Occupation/Topics of Research
Favored Data Sources
Suggested Search Queries
• Internet Search
• Local File Search
• Gem Object Search
• System User Search
Submit Queries/Evaluate Results.
1360
Create gem data objects out of results.
1370
Place objects in container within users database.
1380

FIGURE 13

Root

Examples

```
1401  <Root vers="1.0" id="218838" name="Examples" mime="application/x-gemteq-root">
1402          <rootName>Examples for example</rootName>
1403          <comments>Various things to play with that show a gem server in action
1404  </comments>
1405          <creationDate>11 Aug 1999 14:18 </creationDate>
1406          <creator>
1407                  <creatorName>James Bondo</creatorName>
1408                  <creatorID>jbondo</creatorID>
1409          </creator>
1410          <server>gemserver.gemteq.com</server>
1411          <modificationDate>13 Aug 1999 12:03</modificationDate>
1412          <defaultBib
1413  format="//bibserver.gemteq.com/BibliographyFormats/GenericBook" />
1414          <objList>
1415                  <!-- all are mime: application/x-gemteq-chest by definition --!>
1416
1417          <obj id="210243234" name="ScaryBooks" >
1418                  <containerName>Scary Books</containerName>
1419          </obj>
1420                  <obj id="2832834" name="Various" />
1421                          <containerName>Various</containerName>
1422          </obj>
1423          <obj id="2198348" name="SallyHome" />
1424                          <containerName>Sally's Home Chest</containerName>
1425          </obj>
1426          </objList>
1427  </Root>
```

FIGURE 14

Container

ScaryBooks

```
1501  <Container vers="1.0" id="23444543234" name="ScaryBooks" mime="application/x-gemteq-chest">
1502        <containerName>Some Scary Books</containerName>
1503        <comments>A collection of scary books</comments>
1504        <creationDate>11 Aug 1999 14:19 </creationDate>
1505        <creator>
1506              <creatorName>James Bondo</creatorName>
1507              <creatorID>jbondo</creatorID>
1508        </creator>
1509        <container>//gemserver.gemteq.com/Examples</container>
1510        <modificationDate>13 Aug 1999 12:03</modificationDate>
1511        <defaultBib format="//bibserver.gemteq.com/BibliographyFormats/GenericBook" />
1512        <objList>
1513            <obj id="210243234" name="StephanKing" mime="application/x-gemteq-tray">
1514                <itemName> Misc </itemName>
1515            </obj>
1516            <obj id="218388848" name="DeanKoontz" mime="application/x-gemteq-tray">
1517                <itemName>Dean Koontz</itemName>
1518            </obj>
1519
1520
1521        </objList>
1522  </Container>
```

FIGURE 15

Container

StephanKing

```
1601  <Container vers="1.0" id="210243234" name="StephanKing" mime="application/x-gemteq-tray">
1602        <containerName>Stephan King</containerName>
1603        <comments>A collection of books by Stephan King</comments>
1604        <creationDate>11 Aug 1999 14:19 </creationDate>
1605        <creator>
1606              <creatorName>James Bondo</creatorName>
1607              <creatorID>jbondo</creatorID>
1608        </creator>
1609        <container>//gemserver.gemteq.com/Examples/ScaryBooks</container>
1610        <modificationDate>13 Aug 1999 12:03 </modificationDate>
1611        <defaultBib format="//bibserver.gemteq.com/BibilographyFormats/GenericBook" />
1612        <objList>
1613           <obj id="120983838" name="Carrie" mime="application/x-gemteq-textgem">
1614              <itemName>Carrie</itemName>
1615           </obj>
1616           <obj id="65683838" name="TheStand" mime="application/x-gemteq-textgem">
1617              <itemName>The Stand</itemName>
1618           </obj>
1619
1620           <obj id="777983838" name="IT" mime="application/x-gemteq-textgem">
1621              <itemName>IT</itemName>
1622           </obj>
1623           <obj id="76883838" name=" PetSeminary " mime="application/x-gemteq-textgem">
1624              <itemName>Pet Seminary</itemName>
1625           </obj>
1626           <obj id="120983838" name="TheLangoliers" mime="application/x-gemteq-textgem">
1627              <itemName> The Langoliers </itemName>
1628           </obj>
1629           <obj id="298948393" name="MiscNotes" mime="application/x-gemteq-tray">
1630              <itemName>Misc Notes</itemName>
1631           </obj>
1632           <obj id="298948393" name="Pic" mime="application/x-gemteq-imagegem">
1633              <itemName>Picture of Stephan King</itemName>
1634           </obj>
1635
1636        </objList>
1637  </Container>
```

FIGURE 16

Item

Carrie

```
1701  <Item vers="1.0" id="120983838" name="Carrie" mime="application/x-gemteq-textgem">
1702        <itemName>Carrie</itemName>
1703        <comments>Psi and teenage angst. </comments>
1704        <creationDate> 12 Aug 1999 15:20 </creationDate>
1705        <creator>
1706              <creatorName>Billy Joe Jim Bob</creatorName>
1707              <creatorID>bjjbob</ceatorID>
1708        </creator>
1709        <container>//gemserver.gemteq.com/Examples/ScaryBooks/StephanKing</container>
1710        <modificationDate> 14 Aug 1999 16:20 </modificationDate>
1711        <bib format="//gemserver.gemteq.com/BibilographyFormats/MLABook">
1712              <author>Stephan King</author>
1713              <publicationTitle>Carrie</publicationTitle>
1714              <publisher>Random House</publisher>
1715              <location>New York</location>
1716              <ed> 3 </ed>
1717              <vol>1 </vol>
1718              <date> 2 Aug 1978 </date>
1719              <pageStart>21</pageStart>
1720              <pageEnd>43</pageEnd>
1721              <url>http://www.misclocation.com/some/path/doc.html</url>
1722        </bib>
1723        <!-- an image may have the following attributes
1724            datamime="image/jpg"
1725            compression="application/x-zip"
1726            encoding="application/uuencode"
1727        --!>
1728        <data mime="text/plain" compression="" encoding="">
1729              Carrie was just a normal girl. With an wonderful mother and really good
1730              friends. For some reason she got very angry when they dumped a bucket
1731              of goo upon her at the prom.
1732         </data>
1733  </Item
```

FIGURE 17

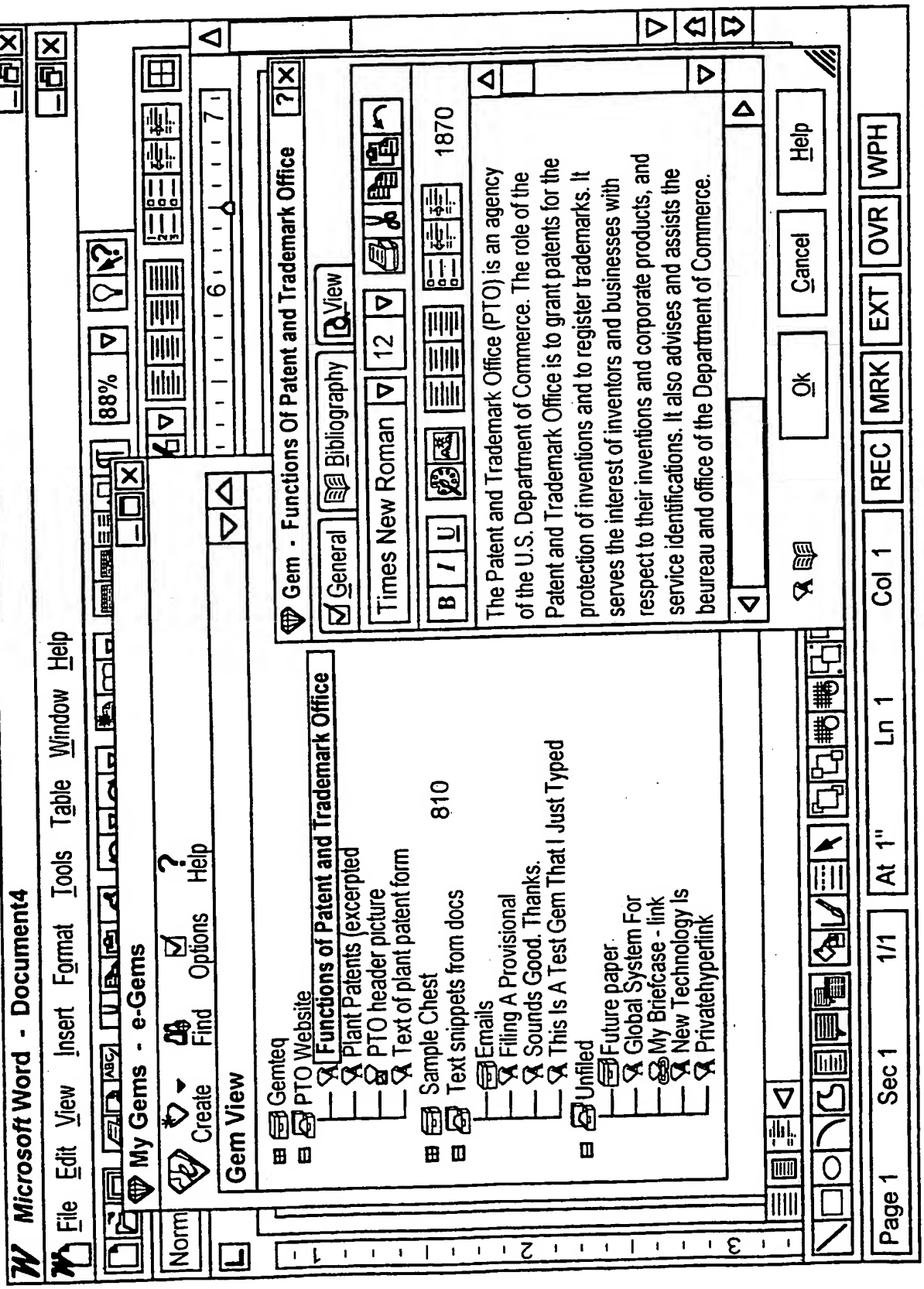

Microsoft Word - Document4
File Edit View Insert Format Tools Table Window Help
My Gems - e-Gems
Create Find Options Help
Gem View
Gemteq
PTO Website
Functions of Patent and Trademark Office
Plant Patents (excerpted
PTO header picture
Text of plant patent form
Sample Chest
Text snippets from docs
Emails
Filing A Provisional
Sounds Good. Thanks.
This Is A Test Gem That I Just Typed
Unfiled
Future paper
Global System For
My Briefcase - link
New Technology Is
Privatehyperlink
Gem - Functions Of Patent and Trademark Office
General
Bibliography
View
Times New Roman
12
B I U
1870
The Patent and Trademark Office (PTO) is an agency of the U.S. Department of Commerce. The role of the Patent and Trademark Office is to grant patents for the protection of inventions and to register trademarks. It serves the interest of inventors and businesses with respect to their inventions and corporate products, and service identifications. It also advises and assists the beureau and office of the Department of Commerce.
Ok
Cancel
Help
Page 1 Sec 1 1/1 At 1" Ln 1 Col 1 REC MRK EXT OVR WPH
810

FIGURE 18

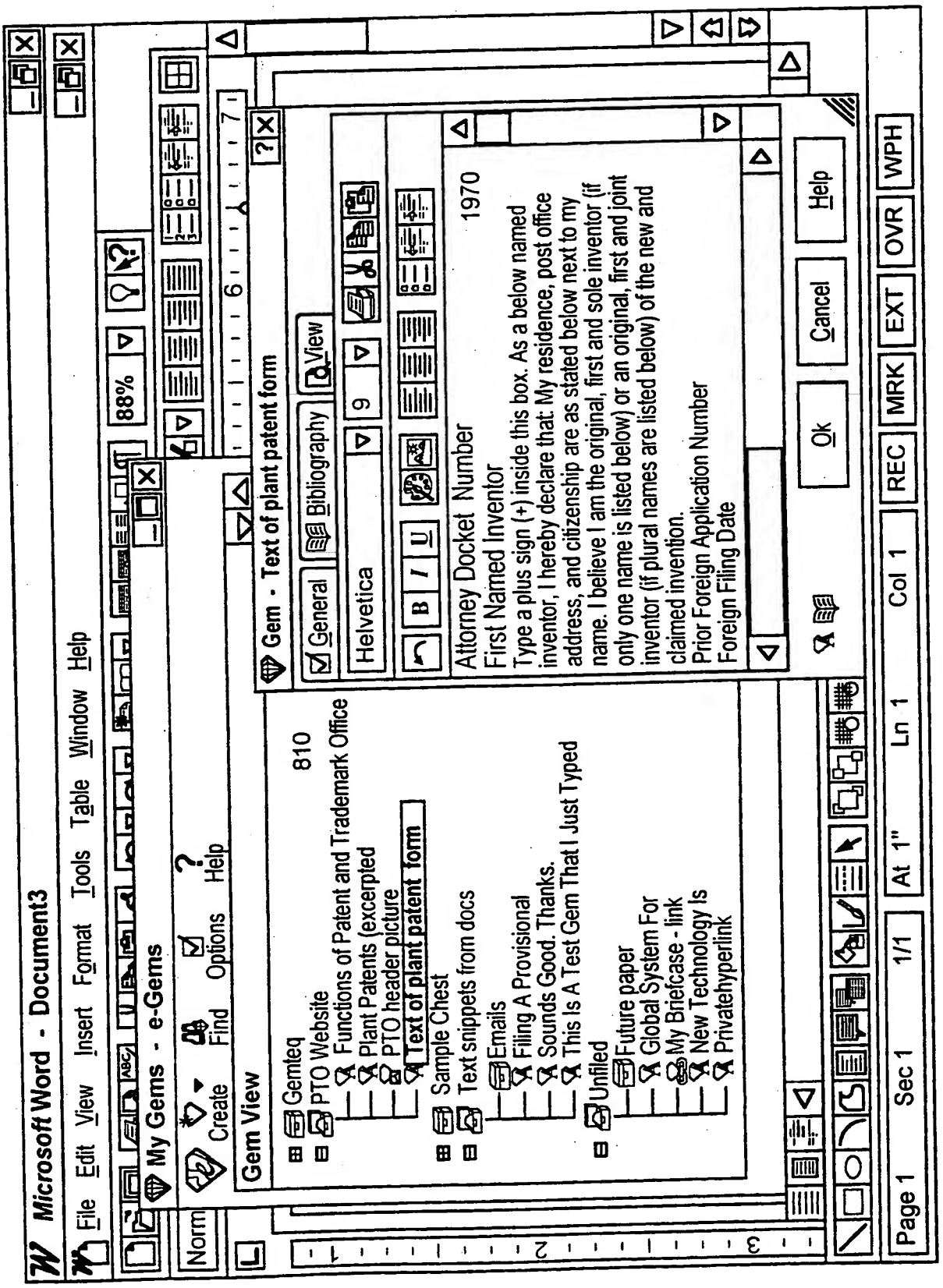

Microsoft Word - Document3
File Edit View Insert Format Tools Table Window Help
My Gems - e-Gems
Create Find Options Help
Gem View
Gemteq 810
PTO Website
Functions of Patent and Trademark Office
Plant Patents (excerpted
PTO header picture
Text of plant patent form
Sample Chest
Text snippets from docs
Emails
Filing A Provisional
Sounds Good. Thanks.
This Is A Test Gem That I Just Typed
Unfiled
Future paper
Global System For
My Briefcase - link
New Technology Is
Privatehyperlink
Gem - Text of plant patent form
General Bibliography View
Helvetica 9
B I U
Attorney Docket Number 1970
First Named Inventor
Type a plus sign (+) inside this box. As a below named inventor, I hereby declare that: My residence, post office address, and citizenship are as stated below next to my name. I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the new and claimed invention.
Prior Foreign Application Number
Foreign Filing Date
Ok Cancel Help
Page 1 Sec 1 1/1 At 1" Ln 1 Col 1 REC MRK EXT OVR WPH

FIGURE 19

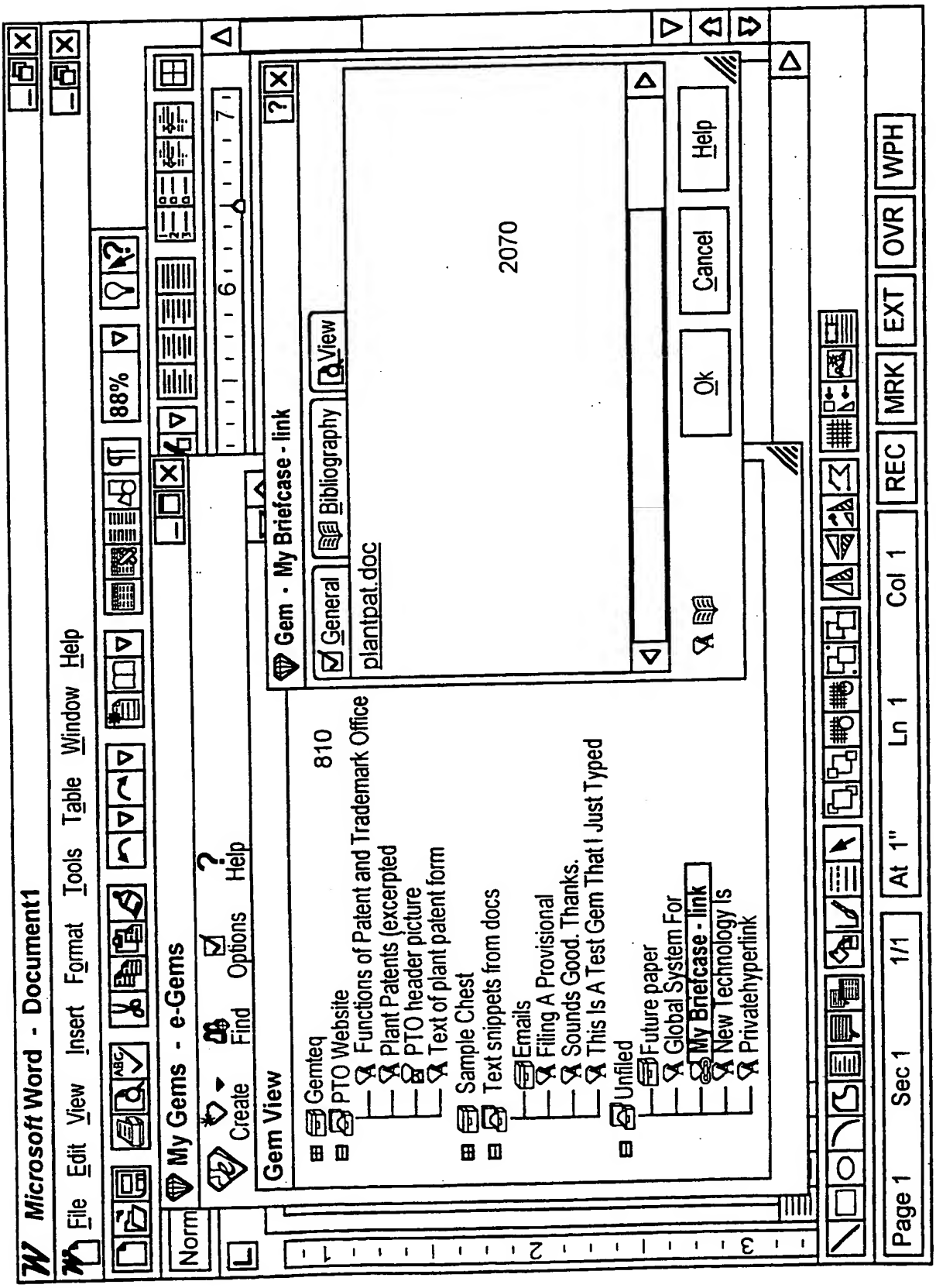

Microsoft Word - Document1
File Edit View Insert Format Tools Table Window Help
My Gems - e-Gems
Create Find Options Help
Gem View
Gemteq
PTO Website
Functions of Patent and Trademark Office
Plant Patents (excerpted
PTO header picture
Text of plant patent form
Sample Chest
Text snippets from docs
Emails
Filing A Provisional
Sounds Good. Thanks.
This Is A Test Gem That I Just Typed
Unfiled
Future paper
Global System For
My Briefcase - link
New Technology Is
Privatehyperlink
Gem - My Briefcase - link
General
Bibliography
View
plantpat.doc
810
2070
Ok
Cancel
Help
Page 1 Sec 1 1/1 At 1" Ln 1 Col 1 REC MRK EXT OVR WPH

FIGURE 20

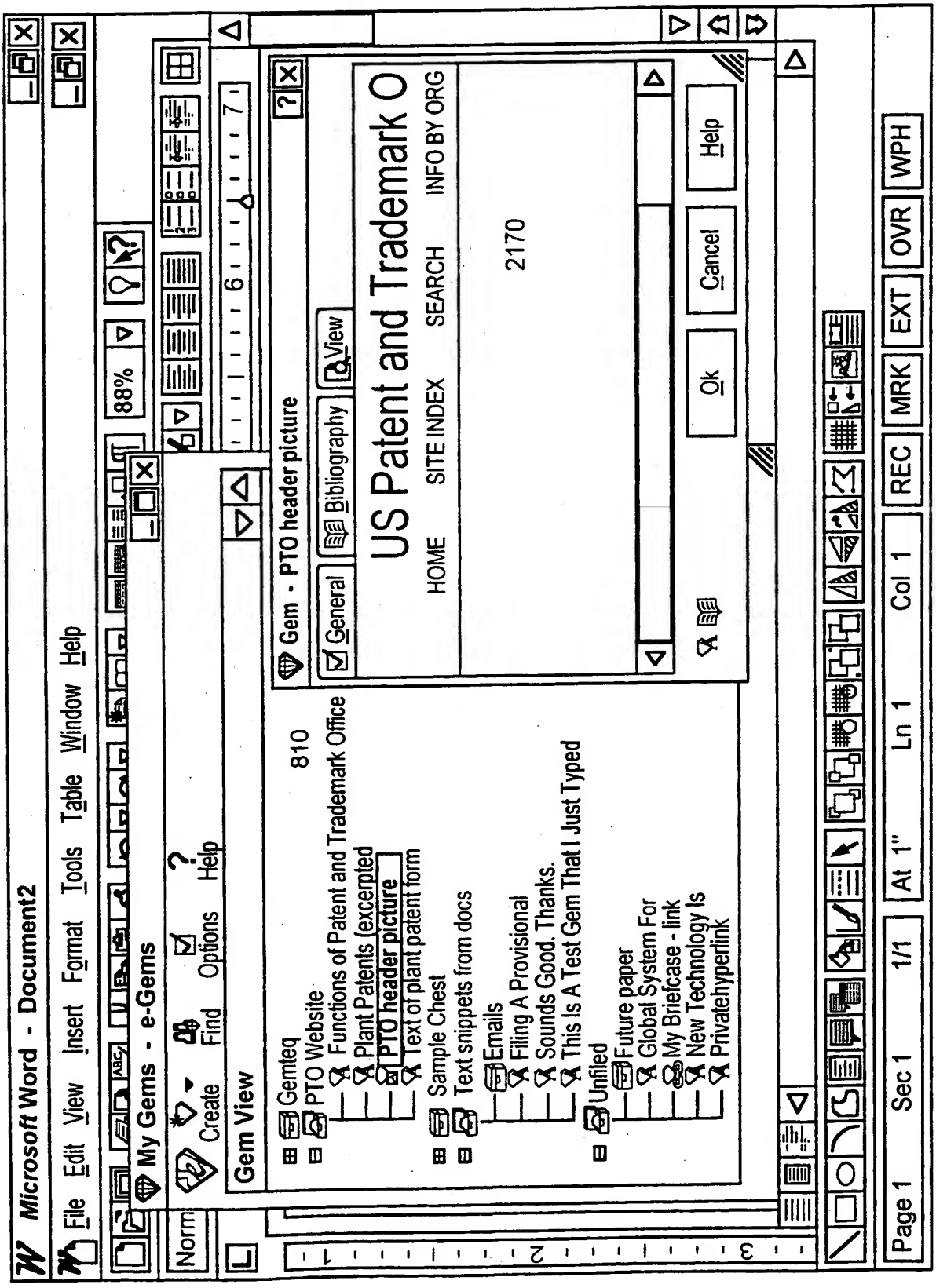

Microsoft Word - Document2
File Edit View Insert Format Tools Table Window Help
My Gems - e-Gems
Create Find Options Help
Gem View
Gemteq
PTO Website
Functions of Patent and Trademark Office
Plant Patents (excerpted
PTO header picture
Text of plant patent form
Sample Chest
Text snippets from docs
Emails
Filing A Provisional
Sounds Good. Thanks.
This Is A Test Gem That I Just Typed
Unfiled
Future paper
Global System For
My Briefcase - link
New Technology Is
Privatehyperlink
810
Gem - PTO header picture
General
Bibliography
View
US Patent and Trademark O
HOME SITE INDEX SEARCH INFO BY ORG
2170
Ok
Cancel
Help
Page 1 Sec 1 1/1 At 1" Ln 1 Col 1 REC MRK EXT OVR WPH

FIGURE 21

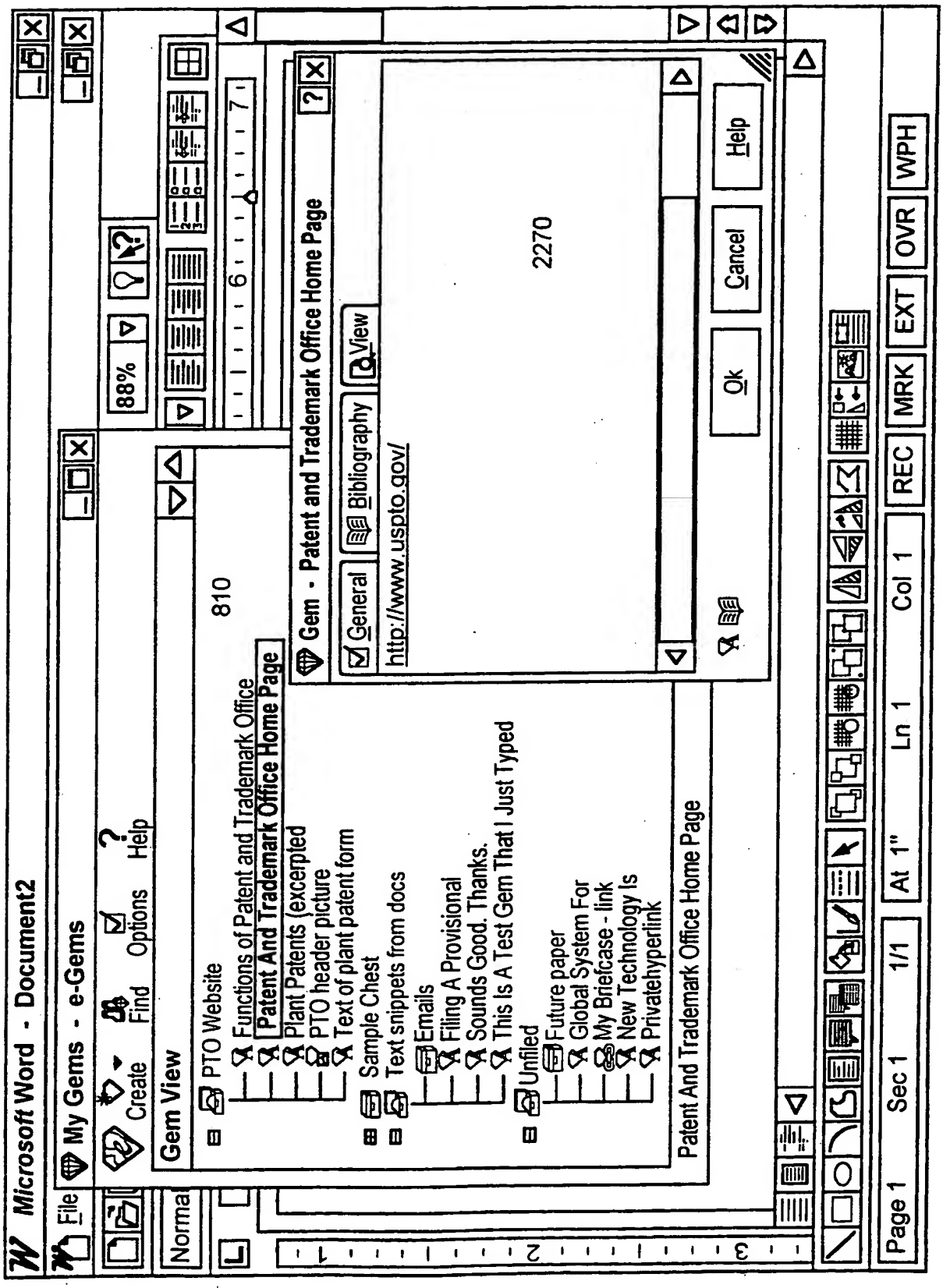

Microsoft Word - Document2
File
My Gems - e-Gems
Create
Find
Options
Help
Gem View
PTO Website
Functions of Patent and Trademark Office
Patent And Trademark Office Home Page
Plant Patents (excerpted
PTO header picture
Text of plant patent form
Sample Chest
Text snippets from docs
Emails
Filing A Provisional
Sounds Good. Thanks.
This Is A Test Gem That I Just Typed
Unfiled
Future paper
Global System For
My Briefcase - link
New Technology Is
Privatehyperlink
Patent And Trademark Office Home Page
810
Gem - Patent and Trademark Office Home Page
General
Bibliography
View
http://www.uspto.gov/
2270
Ok
Cancel
Help
88%
Normal
Page 1
Sec 1
1/1
At 1"
Ln 1
Col 1
REC
MRK
EXT
OVR
WPH

FIGURE 22

INTERNATIONAL SEARCH REPORT

Int. tional Application No
PCT/US 99/30965

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 G06F17/30 G06F17/24

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	EP 0 580 536 A (IBM) 26 January 1994 (1994-01-26) abstract column 1, line 1 -column 4, line 51; figures 1-7	1-19, 41-49, 52-61
X	ANONYMOUS: "Automated Bibliographic Information Gathering and Processing" IBM TECHNICAL DISCLOSURE BULLETIN, vol. 37, no. 4A, 1 April 1994 (1994-04-01), pages 253-254, XP002137312 New York, US the whole document	1,5,41, 42,45, 52,61

-/--

[X] Further documents are listed in the continuation of box C.

[X] Patent family members are listed in annex.

° Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
10 May 2000	23/05/2000

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL – 2280 HV Rijswijk Tel. (+31–70) 340–2040, Tx. 31 651 epo nl, Fax: (+31–70) 340–3016	Deane, E

Form PCT/ISA/210 (second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

Im ational Application No
PCT/US 99/30965

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	ANONYMOUS: "Accessing Remote Data Services Through an Object-Oriented Language." IBM TECHNICAL DISCLOSURE BULLETIN, vol. 34, no. 6, 1 November 1991 (1991-11-01), pages 423-425, XP002137313 New York, US the whole document ---	1,5,41, 42,45, 52,61
A	DATABASE INSPEC 'Online! INSTITUTE OF ELECTRICAL ENGINEERS, STEVENAGE, GB Inspec No. AN 2125759, XP002137314 abstract & ZANIOLO C.: "The Database Language GEM" SIGMOD RECORD, ISSN 0163-5808, vol. 13, no. 4, May 1983 (1983-05), pages 207-218, -----	1,5,41, 42,45, 52,61

1

INTERNATIONAL SEARCH REPORT

Information on patent family members

Int. .tional Application No
PCT/US 99/30965

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP 0580536 A	26-01-1994	JP 2776456 B	16-07-1998
		JP 6096123 A	08-04-1994